

FUJITSU Software Systemwalker Software Configuration Manager

Developer's Guide

Windows/Linux

B1X1-0312-04ENZ0(00)
August 2016

Preface

Purpose of this Document

This document explains how to independently develop the following definitions for managing software parameters using Systemwalker Software Configuration Manager.

- Definition of the source of information collection
- Definition of software information
- Definition of patch information
- Definition of parameter information

Intended Readers

This document is intended for those engaged in developing parameter management definitions for Systemwalker Software Configuration Manager.

It is assumed that readers of this document already have the following knowledge:

- Expertise in the software to be managed
- Expertise in programming

Structure of this Document

The structure of this document is as follows:

[Chapter 1 Definition of Information Collection Sources](#)

This chapter explains how to define the source of information collection.

[Chapter 2 Definition of Software Information](#)

This chapter explains how to define the software information.

[Chapter 3 Definition of Patch Information](#)

This chapter explains how to define the patch information.

[Chapter 4 Definition of Parameter Information](#)

This chapter explains how to define the parameter information.

[Chapter 5 Command Reference](#)

This chapter explains the commands.

[Chapter 6 File Reference](#)

This chapter explains the files.

[Chapter 7 Script Reference](#)

This chapter explains the scripts.

[Chapter 8 Information Collection Source Reference](#)

This chapter explains the source of information collection.

Conventions Used in this Document

Refer to the *Documentation Road Map* for information on the names, abbreviations, and symbols used in this manual.

Abbreviations and Generic Terms Used for Operating Systems

This document uses the following abbreviations and generic terms to indicate operating systems.

Official name	Abbreviation	
Microsoft(R) Windows Server(R) 2012 Datacenter Microsoft(R) Windows Server(R) 2012 Standard	Windows Server 2012	Windows
Microsoft(R) Windows Server(R) 2012 R2 Datacenter Microsoft(R) Windows Server(R) 2012 R2 Standard	Windows Server 2012 R2	
Microsoft(R) Windows Server(R) 2008 Standard Microsoft(R) Windows Server(R) 2008 Standard without Hyper-V Microsoft(R) Windows Server(R) 2008 Enterprise Microsoft(R) Windows Server(R) 2008 Enterprise without Hyper-V	Windows Server 2008	
Microsoft(R) Windows Server(R) 2008 R2 Standard Microsoft(R) Windows Server(R) 2008 R2 Enterprise	Windows Server 2008 R2	
Red Hat(R) Enterprise Linux(R) (for x86)	RHEL (x86)	
Red Hat(R) Enterprise Linux(R) (for Intel64)	RHEL (Intel64)	
Oracle Solaris	Solaris Operating System Solaris OS	Solaris

Export Restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Trademarks

- Adobe, Adobe Reader, and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Interstage, ServerView, and Systemwalker are registered trademarks of Fujitsu Limited.
- Linux is a registered trademark of Linus Torvalds.
- Red Hat, RPM, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.
- Oracle and Java are registered trademarks of Oracle and/or its affiliates in the United States and other countries. Company names and product names used in this document are registered trademarks or trademarks of those companies.
- VMware, the VMware "boxes" logo and design, Virtual SMP, and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.
- Xen and XenSource are trademarks or registered trademarks of XenSource, Inc. in the United States and/or other countries.
- Other company names and product names are trademarks or registered trademarks of their respective owners.

Note that system names and product names in this document are not accompanied by trademark symbols such as (TM) or (R).

Issue Date and Version

Version	Manual code
March 2014: First edition	B1X1-0312-01ENZ0(00) / B1X1-0312-01ENZ2(00)
August 2014: Second edition	B1X1-0312-02ENZ0(00) / B1X1-0312-02ENZ2(00)
November 2015: Third edition	B1X1-0312-03ENZ0(00) / B1X1-0312-03ENZ2(00)

Version	Manual code
August 2016: Forth edition	B1X1-0312-04ENZ0(00) / B1X1-0312-04ENZ2(00)

Copyright

Copyright 2010-2016 FUJITSU LIMITED

Contents

Chapter 1 Definition of Information Collection Sources.....	1
1.1 Definition of Information Collection Sources.....	1
1.1.1 Configuration of Information Collection Sources.....	1
1.1.1.1 Collection Items.....	1
1.1.1.1.1 Multi-Instance.....	1
1.1.1.2 Software Patch Collection Script.....	1
1.1.2 List of Collection Items.....	3
1.2 Management of Information Collection Sources.....	5
Chapter 2 Definition of Software Information.....	7
2.1 Definition of Software Information.....	7
2.2 Definition of Installed Software Information.....	8
Chapter 3 Definition of Patch Information.....	10
3.1 Definition of Patch Information.....	10
3.2 Definition of Applied Patch Information.....	11
Chapter 4 Definition of Parameter Information.....	12
4.1 Definition of Parameters to be Set.....	12
4.1.1 Parameter Settings Definition.....	12
4.1.2 Parameter Types.....	14
4.1.3 Variables that can be Specified as Values.....	14
4.1.4 Parameter Settings Scripts.....	15
4.1.5 Package Files.....	20
4.1.6 Association with the Software.....	21
4.2 Definition of Parameters to be Collected.....	21
4.2.1 Parameter Collection Definition.....	22
4.2.2 Parameter Collection Scripts.....	23
4.2.3 Package Files.....	25
4.2.4 Association with the Software.....	26
Chapter 5 Command Reference.....	28
5.1 Command List.....	28
5.2 Information Collection Source Management Commands.....	29
5.2.1 swcfmg_source_def (Information Collection Source Management Command).....	29
5.3 Software Management Commands.....	34
5.3.1 swcfmg_software (Software Information Management Command).....	34
5.3.2 swcfmg_installedsoftware (Installed Software Information Management Command).....	39
5.4 Patch Management Commands.....	43
5.4.1 swcfmg_patch_def (Patch Information Management Command).....	43
5.4.2 swcfmg_patch_applied (Applied Patch Information Management Command).....	47
5.5 Parameter Management Commands.....	51
5.5.1 swcfmg_param_settingdef (Parameter Settings Definition Management Command).....	51
5.5.2 swcfmg_param_collectingdef (Parameter Collection Definition Management Command).....	56
5.5.3 swcfmg_param_defassoc (Command to Associate Software and Parameter Definitions).....	61
Chapter 6 File Reference.....	65
6.1 File List.....	65
6.2 Definition files of the information collection source.....	65
6.2.1 Information Collection Source File.....	65
6.2.2 Collection Item Information File.....	66
6.2.3 Collection Information CSV File.....	68
6.3 Software Information Definition Files.....	69
6.3.1 Software Information File.....	70
6.3.2 Installed Software Information File.....	72
6.4 Definition files of the patch information.....	73

6.4.1 Patch Information File.....	73
6.4.2 Applied Patch Information File.....	74
6.5 Parameter Information Definition Files.....	75
6.5.1 Parameter Settings Definition File.....	75
6.5.2 Parameter Information File.....	79
6.5.3 Parameter Collection Definition File.....	84
Chapter 7 Script Reference.....	87
7.1 Script List.....	87
7.2 Software Patch Collection Script.....	87
7.2.1 Discovery Script.....	87
7.3 Parameter Settings Script.....	89
7.3.1 Startup Script.....	89
7.4 Parameter Collection Script.....	89
7.4.1 Discovery Script.....	89
Chapter 8 Information Collection Source Reference.....	91
8.1 Information Collection Sources.....	91
8.2 Manual Registration.....	91
8.3 UpdateAdvisor.....	91

Chapter 1 Definition of Information Collection Sources

In order to manage the information of software installed on a server and applied patch information, it is necessary to define the information collection sources. This operation can be performed using only the information collection sources registered in this product.

1.1 Definition of Information Collection Sources

When performing discovery of software patch information, collect the information of the software installed on a server and the applied patch information from the information collection source. The collected information is registered in CMDB (Configuration Management Database) as installed software and applied patches. In this product, the following information collection sources are defined in advance: For details on definition, refer to "[Chapter 8 Information Collection Source Reference](#)".

- Manual registration

The software and patch information manually collected by directly logging in to a server. Using the operation commands of this product, the information can be registered as installed software and applied patches in CMDB. Perform manual registration when managing software and patches which cannot be automatically collected during discovery.

- UpdateAdvisor

UpdateAdvisor (middleware) installed on a business server. During discovery, the Fujitsu middleware information installed on a business server and the applied Fujitsu middleware patch information can be collected.

1.1.1 Configuration of Information Collection Sources

The definition of an information collection source is composed of the following:

- Definition of the collection items

Collection items of the software information and patch information collected by the information collection source.

- Software Patch Collection Script

A script to collect the information of the software installed on a business server and the applied patch information.

1.1.1.1 Collection Items

Define the information collected from an information collection source as the collection items. Define the collection items for each software and patch. Multiple collection items can be defined. For details on the collection items which can be defined, refer to "[1.1.2 List of Collection Items](#)". There are the following categories of collection items.

Identification item

The identification item is the information used to identify a software or patch. Multiple identification items can be defined, and the software and patches whose combinations of collection items are the same are managed as the same software or patch.

Definition item

The definition item is the information used as the definition for a software or patch. The definition information of a software or patch is used as the default value when registering the software information or patch information as an installed software or applied patch in CMDB.

1.1.1.1.1 Multi-Instance

To register the same software as multiple instances of installed software in the same server, use the collection item of the installation location. When installing the same software in different locations, it is registered in CMDB as another installed software.

1.1.1.2 Software Patch Collection Script

The software patch collection script is a script which collects the information of software installed on a business server and the applied patch information. This script is forwarded to the server and runs on it. The information collected from a server is output to the software information CSV file and the patch information CSV file. This file is transferred to the admin server and registered in CMDB as installed software and applied patches.

A software patch collection script is composed of multiple files. In addition, a software patch collection script is created according to the OS type of the target server. The batch file used in Windows and the shell script used in Linux and Solaris are fixed. These files are described below.

- Discovery script

The discovery script is always the first script to be called, and the name of the file is fixed. The name of the batch file is discover.cmd, and the name of the shell script is discover.sh. In this script processing, the software information and patch information collected from a server are always output to the software information CSV file and the patch information CSV file. The script results are also returned. Other CSV files can be created depending on the corresponding information collection sources.

- File attachments

The desired files which can be used with a discovery script. The file and the discovery script will be forwarded to the server. If a file attachment includes a shell script, configure execution privileges within the discovery script of the shell script.

The discovery script outputs the information of the software installed on a server and the applied patch information. A description of this file is shown below.

- Software information CSV

Output the information of the software installed on a server to a collection information CSV file. For details, refer to "[6.2.3 Collection Information CSV File](#)".

The name of this file is fixed. The file name is softwareinfo.csv.

- Patch information CSV

Output the information of patches applied to a server to a collection information CSV file.

The name of this file is fixed. The file name is patchinfo.csv. For details, refer to "[6.2.3 Collection Information CSV File](#)".

Discovery script format (batch files) [Windows]

Create the discovery script for batch files (discover.cmd) using the format below.

The -dir option must be specified as the first argument (%1), and the directory to which the software and patch information should be output must be specified as the second argument (%2). Output the information collected in this directory to the software information CSV file (softwareinfo.csv) and to the patch information CSV file (patchinfo.csv). The script returns 0 if successful, or another value otherwise.

The standard output and standard error output are directed to the agent log.

```
@echo off
setlocal
@rem Process to collect the software information and patch information from a server
<Process of each information collection source>
<Process to output the software and patch information to the %2 directory>
@rem Return results (normal)
if ERRORLEVEL 1 goto ERROR_END
endlocal
exit /B 0
@rem Return results (error)
: ERROR_END
echo ERROR0002 Software and Patch collecting failed. 1>&2
endlocal
exit /B 1
```

Discovery script format (shell scripts) [Linux/Solaris]

Create the discovery script for shell scripts (discover.sh) using the format shown below.

The -dir option must be specified as the first argument (\$1), and the directory to which the software and patch information should be output must be specified as the second argument (\$2) of the shell script. Output the information collected in this directory to the software information CSV file (softwareinfo.csv) and to the patch information CSV file (patchinfo.csv). The script returns 0 if successful, or another value otherwise.

The standard output and standard error output are directed to the agent log.


```
#!/bin/sh
# Process to collect the software information and the patch information from a server
<Process of each information collection source>
<Process to output the software and patch information to the $2 directory>
# Return results
if [ $? = "0" ]; then
# Returns normal
exit 0
else
# Returns normal
echo "ERROR0002 Software and Patch collecting failed." 1>&2
exit 1
fi
```

Output directory for discovery script

When using the discovery script to output software and patch information, output using the directory format shown below. A software information CSV file is created only when the information of the software installed on a server can be collected. A patch information CSV file is created only when the information of patches applied to a server can be collected. The creation of only a patch information CSV file cannot be performed.

```
+ <output directory>
+ Software information CSV file (softwareinfo.csv)
+ Patch information CSV file (patchinfo.csv)
```

Software information CSV file and patch information CSV file format

Create the discovery script so that the software information and the patch information collected using the discovery script can be output to the software information CSV and patch information CSV file. The software information CSV file and patch information CSV file are in the format of a collection information CSV file. For details, refer to "[6.2.3 Collection Information CSV File](#)".

Script package

A script package is a file containing multiple software parameter collection scripts compressed in a ZIP file. This file is used when defining the information collection source. This file is forwarded to the server and unpacked on it.

This ZIP file must be stored using the structure shown below. Place a software patch collection script for use with each OS type under directories with names corresponding to each OS type. It is not necessary to place a software patch collection script for the OS types of servers of which information cannot be collected from the information collection source. Do not create an OS type directory for a software patch collection script of OS types whose data cannot be collected. Place the software patch collection script for at least one OS type. This ZIP file must be stored using the structure shown below. The file attachment can contain directories.

```
<ZIP file root>
+ Windows (*)
+ Discovery script (discover.cmd)
+ File attachment
+ Linux (*)
+ Discovery script (discover.sh)
+ File attachment
+ Solaris (*)
+ Discovery script (discover.sh)
+ File attachment
```

* Note: A directory.

1.1.2 List of Collection Items

The collection items which can be defined are as described below.

Software

Collection Item	Meaning	Value	Remarks
softwareName	Software name	Example: ABCDE-Software	
version	Version	Example: 1.0	
edition	Edition	Example: Standard	
release	Release	Example: 1.0	
architecture	Architecture	Example: x64	
branch	Branch	Example: 10	
license	License	Example: GPLv2	
url	URL	Example: http://ABCDCCompany	
osType	OS type	Example: Windows	When the OS type of the target server is Windows, "Windows" or "Windows EM64T" can be specified. For Linux, "Linux" or "Linux EM64T" can be specified. For Solaris, "Solaris" can be specified.
vendor	Vendor	Example: ABCDE Company	
installDate	Installation date	Example: 2016-01-10 11:11	Use the following format: YYYY-MM-DD HH:MM (local time)
installPath	Installation location	Example: C:\work	<ul style="list-style-type: none"> - Specify the full directory path of the installation location. - When the OS type of the server in which software is installed is Windows, the path delimiter is "\". When the OS type is Linux or Solaris, the path delimiter is "/". - Do not specify a path delimiter at the end of the directory path.
size	Size	Example: 30MB	Specify the size with its unit.
type	Type	Example: Driver utility, etc.	
title	Title	Example: ABCDE-Software.	
description	Description	Example: ABCDE-Software.	

Patch

Item	Meaning	Value	Remarks
patchName	Patch name	Example: 123456	
software	Target software	Refer to the remarks	<p>The software to which the patch is applied. For details on specification, refer to the following:</p> <ul style="list-style-type: none"> - "5.4.1 swcfmg_patch_def (Patch Information Management Command)" - "Software Patch Collection Script" in "Script Reference"
softwareName	Software name	Example: ABCDE-Software 1.0	The name of the software to which the patch is applied. Any desired name can be used.

Item	Meaning	Value	Remarks
			When no value is specified, the software name of the target software is specified.
osType	OS type	Example: Windows	When the OS type of the software to be patched is Windows, "Windows" or "Windows EM64T" can be specified. For Linux, "Linux" or "Linux EM64T" can be specified. For Solaris, "Solaris" can be specified.
significance	Importance	Example: 1	Specify the following values: - 1: Recommended - 2: Important - 3: Security When no value is specified, "1" is configured.
applyDate	Application date	Example: 2016-01-11 11:11	Use the following format: YYYY-MM-DD HH:MM (local time)
releaseDate	Release date	Example: 2016-01-11 11:11	Use the following format: YYYY-MM-DD HH:MM (local time)
title	Title	Example: 123456.	
description	Description	Example: 123456.	

1.2 Management of Information Collection Sources

Use the Information Collection Source Management Command (`swcfmg_source_def`) to define an information collection source.

The functions of the Information Collection Source Management Command are as described below. Refer to "[5.2.1 swcfmg_source_def \(Information Collection Source Management Command\)](#)" for details on the command.

- Registration of an information collection source
- Update of a collection item
- Batch updates of information collection sources
- Deletion of an information collection source
- List display of information collection sources
- Detailed display of a collection item
- Upload of a script package
- Download of a script package

Registration of an information collection source

Register the information collection source. The following items are required to register an information collection source.

- Information collection source ID
- Collection items of the software
- Collection items of patches
- Script package

When applied patch information cannot be collected from an information collection source, patch collection items are not necessary.

Specify a collection item using a CSV file. For details on the file format, refer to "[6.2.2 Collection Item Information File](#)".

For details on script packages, refer to "[1.1.1.2 Software Patch Collection Script](#)" in "[1.1.1 Configuration of Information Collection Sources](#)".

Update of a collection item

Update the collection items of the software or patch.

The collection items of the information collection source of UpdateAdvisor cannot be updated.

Specify a collection item using a CSV file. For details on the file format, refer to "[6.2.2 Collection Item Information File](#)".

Batch update of an information collection sources

Set the information collection sources to valid or invalid.

Specify valid or invalid in the CSV file. For details on the file format, refer to "[6.2.1 Information Collection Source File](#)".

Deletion of an information collection source

Delete an information collection source.

The installed software and applied patch information collected from the deleted information collection source are deleted from CMDB when discovery is performed after the deletion.

Detailed display of a collection item

Display the collection items of the software and patches.

The collection items are output in a CSV file. For details on the file format, refer to "[6.2.2 Collection Item Information File](#)".

Upload of a script package

Update a script package.

The script packages of manually registered information collection sources and UpdateAdvisor information collection source cannot be updated.

For details on script packages, refer to "[1.1.1.2 Software Patch Collection Script](#)" in "[1.1.1 Configuration of Information Collection Sources](#)".

Download of a script package

Download a script package.

The script packages of manually registered information collection sources cannot be downloaded.

A script package is output as a ZIP file. Refer to "[1.1.1.2 Software Patch Collection Script](#)" in "[1.1.1 Configuration of Information Collection Sources](#)".

Deletion of an information collection source

Delete an information collection source.

The software and patch information collected by the deleted information collection source are deleted when discovery is performed after the deletion.

Manually registered information collection sources and the UpdateAdvisor information collection source cannot be deleted.

List display of information collection sources

Display a list of the registered information collection sources.

Chapter 2 Definition of Software Information

This chapter explains how to define software information. To perform the following operations, it is necessary to define software information in advance.

- When registering software as installed software using the Installed Software Command (swcfmg_installedsoftware)
- When defining patch information
- When managing the parameters of software
- When performing discovery of software patch information or configuring the default value to register software as installed software using the Installed Software Command (swcfmg_installedsoftware)

2.1 Definition of Software Information

Use swcfmg_software (Software Information Management Command) to define software information and perform the operations below (refer to "[Chapter 5 Command Reference](#)" for information on the commands):

- Registration of software information
- Update of software information
- Deletion of software information
- Output of software information

Registration of software information

Define the software information. When the information collection source is software which was not manually registered, the software information collected during the discovery of software patch information is automatically registered. In such cases, it is not necessary to register the software information using the Software Information Management Command (swcfmg_software).

When installing this product, the following information can be registered for software whose information collection source is manually registered.

[Registration information]

- Software ID
- Software name
- Version
- OS type
- Vendor

Register a unique entry of information in the format "Software name + Version + OS type". Software with identical information are recognized as being the same, and cannot be registered in the same system. To register software information, specify the CSV file using the Software Information Management Command (swcfmg_software). For details on the file format, refer to "[6.3.1 Software Information File](#)".

Update of software information

Update the registered software information.

To update software information, specify the CSV file using the Software Information Management Command (swcfmg_software). For details on the file format, refer to "[6.3.1 Software Information File](#)".

The "software ID" automatically assigned during registration is used as the key when updating.

When updating information, we recommend revising files generated using the software information output feature, and deleting software information that does not require update. Use these files as the input for the command to update information.

Note that an update error will occur if no software ID, or an incorrect one, is entered.

Deletion of software information

Delete the registered software information.

Specify the software ID of the software information to be deleted.

Check the software ID using the software information output function.

Note that software information cannot be deleted for software that has registered installed software information. Ensure that installed software information has been deleted before deleting the software information that is not required.

Output of software information

Output the registered software information as a software information CSV file.

The output destination is the standard output of the command or the specified file.

2.2 Definition of Installed Software Information

Register installed software information in order to manage software that cannot be automatically collected using discovery or to configure the software parameters.

Use `swcimg_installedsoftware` (Installed Software Information Management Command) to register installed software information. Perform the operations below (refer to "[Chapter 5 Command Reference](#)" for information on the command):

- Registration of installed software information
- Deletion of installed software information
- Output of installed software information

Registration of installed software information

Register the installed software information for the server on which software registered using the Software Information Management Command was installed.

[Registration information]

- Server ID or IP address
- Software ID

For software that cannot be automatically collected using discovery, register the installation status of each server using the Installed Software Information Management Command.

Information about installed software is registered using the combination of the server ID or IP address of the managed server and the software ID.

Information about installed software is registered using the combination of the IP address of the managed server and the software ID.

Specify CSV files in the format below using the Installed Software Information Management Command (refer to "[Chapter 6 File Reference](#)" for information on file format):

```
[#]<server>,<software-id>  
...
```

Check the software ID using the software information output function. When registering the information of a multi-instance installed software, specify the software IDs and installation locations separated by semicolons (";") for software-id.

An error will occur if Systemwalker Software Configuration Manager does not manage the server or if the software is not registered in it.

An error will also occur if the specified combination of server and software ID already exists.

Deletion of installed software information

Delete the installed software information that was registered using the Installed Software Information Management Command.

To delete installed software information, specify CSV files in the format above using the Installed Software Information Management Command.

An error will occur if Systemwalker Software Configuration Manager does not manage the server or if the software is not registered in it. Note that an error will also occur if the specified combination of server and software contains a server that already exists or if the software has not been configured as installed software.

Output of installed software information

Output the installed software information in the CSV format below (the output destination is the standard output of the command or the specified file):

```
#IP address,Software ID
<IP address>,<Software ID>
...
```

Specify the -all option to output information registered in the discovery process as well as installed software information that was registered using this command. Refer to "[Chapter 5 Command Reference](#)" for information on options when executing commands.

Chapter 3 Definition of Patch Information

This chapter explains how to define patch information. To perform the following operations, it is necessary to define patch information in advance.

- When registering a patch as an applied patch using the Applied Patch Command (swcfmg_patch_applied)
- When performing discovery of software patch information or configuring a default value while registering a patch as an applied patch using the Applied Patch Command (swcfmg_patch_applied)

3.1 Definition of Patch Information

Use the Patch Information Management Command (swcfmg_patch_def) to define patch information.

The functions of the Patch Information Management Command are described below. Refer to "[5.4.1 swcfmg_patch_def \(Patch Information Management Command\)](#)" for information on the command.

- Registration of patch information
- Update of patch information
- Deletion of patch information
- Output of patch information

Registration of patch information

Define the patch information. When the information collection source is a patch which was not manually registered, the patch information collected during the discovery of software patch information is automatically registered. In such cases, it is not necessary to register the patch information using the Patch Information Management Command (swcfmg_patch_def). When the information collection source is a patch for UpdateAdvisor, it is necessary to register a patch in this product using the Fujitsu Middleware Released Patch Registration Command (swcfmg_fjmwpatch_update) to register the patch information. For details, refer to "[swcfmg_fjmwpatch_update \(Fujitsu Middleware Released Patch Registration Command\)](#)" in the *Reference Guide*.

To register patch information, it is necessary to define the information of the software to be patched in advance. For details on how to define software information, refer to "[Chapter 2 Definition of Software Information](#)".

To register patch information, specify the CSV file using the Patch Information Management Command. For details on the file format, refer to "[6.4.1 Patch Information File](#)".

Update of patch information

Update the registered patch information.

To update the patch information, specify the CSV file using the Patch Information Management Command. For details on the file format, refer to "[6.4.1 Patch Information File](#)".

Deletion of patch information

Delete the registered patch information.

Specify the patch ID of the patch information to delete. Confirm the patch ID using the patch information output function.

Note that patch information cannot be deleted when applied patch information is registered for the patch information. When deleting unnecessary patch information, delete applied patch information in advance.

Output of patch information

Output the registered patch information as a patch information CSV file.

The output destination is the standard output of the command or the specified file.

3.2 Definition of Applied Patch Information

Register applied patches in order to manage patches which cannot be automatically collected using discovery as applied patches.

Use the Applied Patch Information Management Command (swcfmg_patch_applied) to register applied patch information.

The functions of the Applied Patch Information Management Command are described below. Refer to "[Chapter 5 Command Reference](#)" for information on the command.

- Registration of applied patch information
- Deletion of applied patch information
- Output of applied patch information

Registration of applied patch information

Register applied patch information for servers which have been patched with patches registered using the Patch Information Management Command.

[Registration information]

- Server ID or IP address
- Software ID
- Patch ID

For patches which cannot be automatically collected using discovery, use the Applied Patch Information Management Commands to register the application status for each server.

Applied patch information is registered using the combination of the server ID or IP address of the managed server, the software ID, and the patch ID.

Specify CSV files in the format below using the Applied Patch Information Management Command. Refer to "[Chapter 6 File Reference](#)" for information on file format.

```
[#]<server>,<software-id>,<patch-id>  
...
```

Confirm the patch ID using the patch information output function. When registering an applied patch for a multi-instance installed software, specify the software IDs and installation locations separated by semicolons (";") for software-id.

An error will occur if Systemwalker Software Configuration Manager does not manage the server or if the software or patch is not registered in it.

An error will also occur if the specified combination of server, software, and patch already exists.

Deletion of applied patch information

Delete registered applied patch information using the Applied Patch Information Management Command.

To delete applied patch information, specify a CSV file in the above format using the Applied Patch Information Management Command.

An error will occur if Systemwalker Software Configuration Manager does not manage the server or if the software or patch is not registered in it.

Output of applied patch information

The applied patch information file is output as a CSV file in the above format.

The output destination is the standard output of the command or the specified file.

Chapter 4 Definition of Parameter Information

The following definitions are required when managing software parameters. Note that these definitions are not required if using the software definitions pre-registered in this product.

- Definition of Parameters to be Set
- Definition of Parameters to be Collected

The settings below are then configured based on these definitions.

- Parameter Value Settings

Refer to the *Operation Guide* for information on how to set parameter values.



See

There are two types of software definitions:

- a. Definitions registered in this product
- b. User-created definitions

Refer to "Middleware List" in the *Parameter Reference* for information on a.

4.1 Definition of Parameters to be Set

Create a parameter settings definition for defining parameters that can be configured in the software. Use the parameter settings definition management command (`swcfmg_param_settingdef`) to create a parameter settings definition. To associate the software and parameter settings definitions, use the command to associate software and parameter definitions (`swcfmg_param_defassoc`). Refer to "[Chapter 5 Command Reference](#)" for information on the commands.

The knowledge required to create this parameter settings definition is explained below:

- Parameter settings definition
Specify the list of parameters (key name, type, and default value), method, and script package for this parameter settings definition.
- Parameter types
The parameter type can be specified as boolean, number, string, string array, or map.
- Variables that can be specified as values
Variables can also be specified as the default value.
- Parameter settings scripts
Parameters are configured in the software using a parameter settings script. This script is forwarded to the server and runs on it.
- Package files
The parameter settings script is converted to a package file compressed in ZIP format and then associated with the parameter settings definition.
- Association with the software
Although the parameter settings definition defines the parameters that can be configured in the software, it does not specify the corresponding software. For this reason, the parameter settings definition must be associated with the software information.

4.1.1 Parameter Settings Definition

The parameter settings definition is used to define the parameters that can be configured in the software. Specify the list of parameters (key name, type, and default value), method, and script package for this definition.

In the parameter list, specify multiple parameters that can be configured in the software. Each parameter is represented by a key name, type, and default value. However, the default value can also be omitted.

For the parameter setup method, specify which method should be used to configure the parameter values in the software. Select either a batch file parameter settings script that can be used in Windows, or a shell script parameter settings script that can be used in Linux as the method.

A script package is a package file containing the parameter settings script compressed in ZIP format.

Required information

The information to be specified in the parameter settings definition is shown below.

Tag name	Allowable range	Description	Mandatory	Settings
name	256 characters or less	Specifies the parameter settings definition name.	Y	
description	256 characters or less	Specifies the parameter settings definition description.	Y	
method	Select an option	Specifies the parameter setup method script.	Y	Select from the following options: <ul style="list-style-type: none"> - "cmd": Calls the startup script (startup.cmd). - "sh": Calls the startup script (startup.sh).
parameters	1 or more	Specifies multiple parameters that can be configured in the software.	Y	
key	1 to 256 bytes	Specifies the parameter key.	Y	Characters that can be used are alphanumeric characters, ".", "_", and "-". However, the first character must only be alphabetic.
	Select an option	Specifies the value type.	Y	Select from the following options: <ul style="list-style-type: none"> - "boolean": true, false - "number": Number - "string": Character string - "string array": Character string array - "map": (Map) Refer to " 4.1.2 Parameter Types " for details.
	4096 characters or less	Specifies the default value if a value must be set. The value can be changed when configuring this parameter, but it cannot be left blank.	N	Values that can be specified are determined by 'type'. The string "__EMPTY__" (prefixed and suffixed by 2 underscores) cannot be specified. Variables can be specified as values by prefixing them with # (to specify # or \ as part of the value, prefix them with the \ escape character). Refer to " 4.1.3

Tag name	Allowable range	Description	Mandatory	Settings
				Variables that can be Specified as Values for details.
label	64 characters or less	Specifies the label used to display the parameter in the window.	N	
description	256 characters or less	Specifies the parameter description.	N	
Script package	2 MB or less	Specifies the package file containing the parameter settings script compressed in ZIP format.	Y	For details on parameter settings scripts, refer to " 4.1.4 Parameter Settings Scripts ". For details on package files, refer to " 4.1.5 Package Files ".

4.1.2 Parameter Types

A type can be specified for parameters in a parameter settings definition or a parameter collection definition. The following types can be specified:

- boolean

The values true and false can be specified.

- number

Numeric values between -2,147,483,648 and 2,147,483,647 can be specified.

- string

Strings (including empty strings) can be specified.

- string array

String arrays with zero or more elements can be specified (the array index starts from 1).

- map

Zero or more pairs (entries) of subkeys and values can be specified as values. Subkeys can contain alphanumeric characters, periods (.), underscores (_), and hyphens (-).

Example: Specifying values for three subkeys:

```
Subkey: subkey1, value: data1
Subkey: subkey2, value: data2
Subkey: subkey3, value: data3
```

4.1.3 Variables that can be Specified as Values

Variables can be specified as parameter values when specifying a server or OS value (computer name, host name, or IP address) as a parameter value.

Specify variables using the following format:

```
#{<variable name>}
```

Example:

```
<value>#{server.os.computername}</value>
```

List of variables

The variables that can be used are shown below, together with their descriptions:

Variable name	Description
server.os.computername	Computer name/host name
server.nic.ipaddress	IP address of NIC

Note

To specify # or \ as part of the value, prefix them with the \ escape character.

4.1.4 Parameter Settings Scripts

A parameter settings script is a script used to configure parameters in software. This script is forwarded to the server and runs on it.

A parameter settings script is composed of multiple files. Note that the names of Windows batch files and Linux shell scripts are fixed. These files are described below.

- Startup script

The startup script is always the first script to be called.

The file name of this script is fixed. It is named startup.cmd (if a batch file) or startup.sh (if a shell script). This script must call the environment variable settings script, which is the input, and return the result, which is the output. Other processes can be created for different software products.

- Environment variable settings script

Environment variables are used to input parameter values in the startup script. These environment variables are configured using the environment variable settings script.

This script is named setenv.cmd (if a batch file) or setenv.sh (if a shell script). This product generates the environment variable settings script from the parameters settings definition and parameter information.

- Parameter information XML file for configured parameters

Input to the startup script is also passed via the parameter information XML file.

This file is named parameterinfo.xml. This product generates the parameter information XML file from the parameters settings definition and parameter information.

- File attachments

Any file can be used from the startup script. The file will be forwarded to the server with the startup script.

If a file attachment includes a shell script, configure execution privileges within the startup script of the shell script.

Startup script format (batch files) [Windows]

Create the startup script for batch files (startup.cmd) using the format below.

The environment variable settings script (setenv.cmd) is called first. Specify a process that acquires parameter values from the environment variables and sets the parameters in the software. Parameter values can also be acquired from the parameter information XML file. The script returns 0 if successful, or another value otherwise. The standard output and standard error output are directed to the agent log.

```
@echo off
setlocal
@rem Configure environment variables
call .\setenv.cmd
@rem Software setup process
<Processing for each software product>
setup -host %manager_name%
@rem Return results (normal)
if ERRORLEVEL 1 goto ERROR_END
endlocal
exit /B 0
@rem Return results (error)
: ERROR_END
```

```
echo ERROR0001 Parameter settings failed. 1>&2
endlocal
exit /B 1
```

Startup script format (shell scripts) [Linux]

Create the startup script for shell scripts (startup.sh) using the format below.

The environment variable settings script (setenv.sh) is called first. Specify a process that acquires parameter values from the environment variables and sets the parameters in the software. Parameter values can also be acquired from the parameter information XML file. The script returns 0 if successful, or another value otherwise. The standard output and standard error output are directed to the agent log.

```
#!/bin/sh
# Configure environment variables
source ./setenv.sh
# Software setup process
<Processing for each software product>
setup -host ${manager_name}
# Return results
if [ $? = "0" ]; then
# Returns normal
exit 0
else
# Returns an error
echo "ERROR0001 Parameter settings failed." 1>&2
exit 1
fi
```

Environment variable settings script format (batch files) [Windows]

This product generates the environment variable settings script batch file (setenv.cmd) in the format below.

The environment variable names are determined by the parameter keys in the parameter settings definition (parameter information). The environment variable values are determined by the parameter values in the parameter settings definition or in the parameter information. However, the following conversions are performed:

- Periods (.) and hyphens (-) in keys are converted to underscores (_)

Environment variable names are converted to strings in which periods (.) and hyphens (-) in keys are converted to underscores (_).

- String array-type environment variables combine the key index with the array index

An environment variable is created for each element of the array, with the name being the key, followed by underscore, followed by the index. The value of the environment variable name containing the key is the number of elements in the array. If the array has no elements, an environment variable name will be created for the key only. The index starts from 1.

- Map-type environment variables combine the key and subkey

An environment variable is created for each subkey, with the name being the key, followed by underscore, followed by the subkey. The value of the environment variable name containing the key is the number of entries in the map. If the map has no entries, an environment variable name will be created for the key only.

- Specify __EMPTY__ for empty values

```
set <parm key1>=<parm val >
set <parm key array>=<num of elements>      ...string array type
set <parm key array>_<index> =<parm val >    ...string array type
set <parm key map>=<num of entries>         ...map type
set <parm key map>_<subkey>=<parm val >     ...map type
<Define as many environment variables as there are parameters>
```

Example:

```
set manager_name=server1
set manager_adata=2          ...string array type
set manager_adata_1=data1   ...string array type
```

```

set manager_adata_2=data2          ...string array type
set manager_bdata=0                ...string array type with 0 elements
set manager_cdata=3                ...map type
set manager_cdata_subkey1=data1    ...map type
set manager_cdata_subkey2=data2    ...map type
set manager_cdata_subkey3=data3    ...map type
set manager_ddata=0                ...map type with 0 entries

```

Environment variable script format (shell scripts) [Linux]

This product generates the environment variable settings script shell script (setenv.sh) in the format below.

The environment variable names are determined by the parameter keys in the parameter settings definition (parameter information). The environment variable values are determined by the parameter values in the parameter settings definition or in the parameter information. However, the following conversions are performed:

- Periods (.) and hyphens (-) in keys are converted to underscores (_)

Environment variable names are converted to strings in which periods (.) and hyphens (-) in keys are converted to underscores (_).

- String array-type environment variables combine the key index with the array index

An environment variable is created for each element of the array, with the name being the key, followed by underscore, followed by the index. The value of the environment variable name containing the key is the number of elements in the array. If the array has no elements, an environment variable name will be created for the key only. The index starts from 1.

- Map-type environment variables combine the key and subkey

An environment variable is created for each subkey, with the name being the key, followed by underscore, followed by the subkey. The value of the environment variable name containing the key is the number of entries in the map. If the map has no entries, an environment variable name will be created for the key only.

- Enclose values that contain spaces in double quotation marks ("")

```

<parm key>=<parm val >
<parm key>=<num of elements>    ...string array type
<parm key>_<index>=<parm val >  ...string array type
<parm key>=<num of entries>     ...map type
<parm key>_<subkey>=<parm val > ...map type
<Define as many environmental variables as there are parameters>

```

Example:

```

manager_name=server1
manager_adata=2          ...string array type
manager_adata_1=data1   ...string array type
manager_adata_2=data2   ...string array type
manager_bdata=0         ...string array type with 0 elements
manager_cdata=3         ...string array type
manager_cdata_subkey1=data1 ..map type
manager_cdata_subkey2=data2 ..map type
manager_cdata_subkey3=data3 ..map type
manager_ddata=0         ...map type with 0 entries

```

Parameter information XML file format

The parameter information XML file for configured parameters is generated by this product. Reference the content of the key tags and their values <parameters> section of this file. Other tags are omitted or are not required for referencing.

- Tag to be referenced

parameters

- Tag to be omitted

description

- Tag not required for referencing

name

An example of a generated parameter information XML file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<parameterInfo version="3.0">
  <name>parmInfoName</name>
  <parameters>
    <parameter>
      <key>parmKey</key>
      <value>parmVal</value>
    </parameter>
    ...
  </parameters>
</parameterInfo>
```

An example of a parameter information XML file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<parameterInfo version="3.0">
  <name>Parameter Info</name>
  <parameters>
    <parameter>
      <key>manager.name</key>
      <value>server1</value>
    </parameter>
    <parameter>
      <key>manager.adata</key>
      <array>
        <element>data1</element>
        <element>data2</element>
      </array>
    </parameter>
    <parameter>
      <key>manager.cdata</key>
      <map>
        <entry>
          <subkey>subkey1</subkey>
          <value>data1</value>
        </entry>
        <entry>
          <subkey>subkey2</subkey>
          <value>data2</value>
        </entry>
        <entry>
          <subkey>subkey3</subkey>
          <value>data3</value>
        </entry>
      </map>
    </parameter>
  </parameters>
</parameterInfo>
```



Note

- **Commands that cannot be used with a script**

Do not execute the following commands from a script, because this will cause the script to enter standby status on the business server, and its processing will not complete:

- Commands that require interaction [Windows/Linux]

- Commands for which a window opens during execution [Windows]
- AT commands [Windows]
- Shell scripts created using PowerShell [Windows]
- Commands running in full-screen mode [Linux]

- Parameter settings scripts

- Order of execution

The parameter settings script is operated according to the order specified in the Parameter Settings Wizard.

- Startup script

- Execution privileges

Must be executed by an Administrator in Windows.

Must be executed by a superuser in Linux.

- Current directory

The current directory is the path in which the startup script files are stored.

- Execution privileges [Linux]

Configured automatically during execution.

- Line feed

In Windows, use CR+LF.

In Linux, use LF.

- Byte order mark (BOM) [Linux]

Do not include the UTF-8 byte order mark (BOM) in shell scripts.

- Environment variable settings script

- Environment variable definitions

Environment variables cannot be defined if parameter values are not configured. Accordingly, whether parameter values will be configured in the startup script is determined by whether environment variables have been defined.

The method for determining whether environment variables have been defined is shown below:

Batch files: [Windows]

```
if defined <environment variable> (<defined process>) else <undefined process>
```

Example:

```
set PARAM=%hostname%
if defined parameter_Key1 (set PARAM=-v %parameter_Key1% %PARAM%)
```

Shell scripts: [Linux]

```
${<environment variable>}+${<environment variable>}
```

Example:

```
PARAM=${hostname}
PARAM="${parameter_Key1+"-v ${parameter_Key1}" } ${PARAM}"
```

- Empty strings in values [Windows]

An empty string cannot be configured in environment variables in Windows. For this reason, specify empty strings in batch files using "__EMPTY__" (prefixed and suffixed by 2 underscores). Note that as a result, the string "__EMPTY__" cannot be used as a parameter value.

- Symbols (special characters)

Some parameter values passed by an environment variable settings script do not accept symbols (special characters), or an escape character must be specified before a symbol (special character).

- **File attachments**

- Execution privileges [Linux]

If a file attachment includes a shell script, configure execution privileges within the startup script of the shell script.

4.1.5 Package Files

A package file refers to one or more compressed files in ZIP format. These files are used to associate the parameter settings script with the parameter settings definition and parameter information. Package files are decompressed on the server once they have been forwarded.

The package file to be associated with the parameter settings definition is known as the script package. The package file registered in the parameter information is known as the parameter package. The parameter settings scripts contained in each of these package files is different. An explanation is provided below:

- Script package

Includes the startup script of the parameter settings script, and the file attachment. The startup script must be included.

- Parameter package

Includes the file attachment of the parameter settings script.

Script package

A script package is a ZIP file that contains a startup script and a file attachment. This ZIP file must be stored using the structure shown below. Always store the startup script in the ZIP file root. The file attachment can contain a directory.

```
<ZIP file root>
+ <Startup script (startup.cmd or startup.sh)>
+ <File attachment>
```

Parameter package

A parameter package is a ZIP file that contains a file attachment. This ZIP file must be stored using the structure shown below. The file attachment can contain a directory.

```
<ZIP file root>
+ <File attachment>
```

Execution example of the parameter settings script

Script packages and parameter packages are forwarded to the server and decompressed on it. The environment variable settings script generated by this product is also forwarded and stored in the work directory on the server as described below - note that the directory in which the startup script is stored becomes the current directory, and the startup script is executed.

```
<work directory>
+ <software id>
  + scriptpkg
    | + Startup script (startup.cmd or startup.sh)
    | + Environment variable settings script (setenv.cmd, setenv.sh)
    | + File attachment in the script package
  + paramdata
    | + Parameter information XML file (parameterinfo.xml) for configured parameters
  + parampkg
    + Parameter package attachment
```

A description of each directory is shown below:

- Software ID

Directory unique to each software product, with the name *<software id>*.

- scriptpkg

Stores the startup script, the file attachment (both included in the script package) and the environment variable settings script.

- paramdata

Stores the parameter information XML file (parameterinfo.xml) for configured parameters.

- parampkg

Stores the file attachment included in the parameter package. When referring to this file in the startup script, use the relative path.



- **Package files**

- Store the startup script in the ZIP file root, not in the directory, otherwise it cannot be executed.
- Package files forwarded to the server are deleted once the script has been executed.

4.1.6 Association with the Software

Although the parameter settings definition defines the parameters that can be configured in the software, it does not specify the corresponding software. For this reason, the parameter settings definition must be associated with the software information.

Multiple software information items can be associated for each parameter settings definition. For this reason, parameter settings definitions can be aggregated as a single item, even if there are multiple editions of the software. However, since Windows and Linux editions cannot coexist, distinct parameter settings definitions must be created in such cases. Note that it is also possible to associate multiple parameter settings definitions with software information.

4.2 Definition of Parameters to be Collected

Create a parameter collection definition for defining parameters that can be collected from the software. Use the parameter collection definition management command (swcfmg_param_collectingdef) to create a parameter collection definition. To associate the software and parameter collection definitions, use the command to associate software and parameter definitions (swcfmg_param_defassoc). Refer to "[Chapter 5 Command Reference](#)" for information on the commands.

The knowledge required to create this parameter collection definition is explained below.

- Parameter collection definition

Use the parameter collection definition to define parameters that can be collected from the software. Specify the list of parameters (key name and type), method, and script package for this parameter collection definition.

- Parameter types

The parameter type can be specified as boolean, number, string, string array, or map (refer to "[4.1.2 Parameter Types](#)" for details).

- Parameter collection scripts

Parameters are collected from the software using a parameter collection script. This script is forwarded to the server and runs on it. Information about the parameters collected from the software is output to files such as the parameter information XML file (parameterinfo.xml). This file is transferred to an admin server.

- Package files

The parameter collection script is converted to a package file compressed in ZIP format and then associated with the parameter collection definition.

- Association with the parameter collection definition

Although the parameter collection definition defines the parameters that can be collected from the software, it does not specify the corresponding software. For this reason, the parameter collection definition must be associated with the software information.

If multiple parameter collection definitions are to be associated with a single software information item, you cannot specify a duplicate key for the parameter collection definitions.

4.2.1 Parameter Collection Definition

The parameter collection definition is used to define the parameters that can be collected from the software. Specify the list of parameters (key name and type), method, and script package for this definition.

In the parameter list, specify multiple parameters that can be collected from the software. Each parameter is represented by a key name and type.

For the parameter collection method, specify which method should be used to collect the parameter values from the software. Select either a Windows batch file or Linux shell script.

A script package is a package file containing the parameter collection script compressed in ZIP format.

Specified details

An explanation of details to be specified in the parameter collection definition is shown below.

Tag name	Allowable range	Description	Mandatory	Settings
name	256 characters or less	Specifies the parameter collection name.	Y	
method	Selection	Specifies the parameter collection method.	Y	Select from the following options: <ul style="list-style-type: none"> - "cmd": Calls the 'discover.cmd' discovery script. - "sh": Calls the 'discover.sh' discovery script.
parameters	1 or more	Specifies multiple parameters that can be collected from the software.	Y	
key	1 to 256 bytes	Specifies the parameter key.	Y	Characters that can be used are alphanumeric characters, ".", "_", and "-". However, the first character must only be alphabetic.
type	Select an option	Specifies the value type.	Y	Select one of the following options: <ul style="list-style-type: none"> - "boolean" - "number" - "string" An empty string can also be specified. - "string array" Empty arrays can also be specified. - "map" Empty maps can also be specified. Refer to " 4.1.2 Parameter Types " for details.

Tag name	Allowable range	Description	Mandatory	Settings
label	64 characters of less	Specifies the label used to display the parameter in the window.	N	
Script package	2 MB or less	Specifies the package file containing the parameter collection script compressed in ZIP format.	Y	For details on parameter collection scripts, refer to " 4.2.2 Parameter Collection Scripts ". For details on package files, refer to " 4.1.5 Package Files ".

4.2.2 Parameter Collection Scripts

A parameter collection script is a script used to collect parameters from the software. This script is forwarded to the server and runs on it. Information about the parameters collected from the software is output to files such as the parameter information XML file (parameterinfo.xml). This file is transferred to an admin server.

A parameter collection script is composed of multiple files. Note that the names of Windows batch files and Linux shell scripts are fixed. These files are described below.

- Discovery script

The discovery script is always the first script to be called.

This script is named discover.cmd (if a batch file) or discover.sh (if a shell script). This script must output information about the parameters collected from the software to the parameter information XML file (parameterinfo.xml). Other processes can be created for each different software product.

- File attachments

Any file can be used from the discovery script. The file and the discovery script will be forwarded to the server.

If a file attachment includes a shell script, configure execution privileges within the discovery script of the shell script.

The discovery script outputs information about parameters collected from the software to a file. A description of this file is shown below.

- Collected parameter information XML file

XML file containing information about collected parameters. This file has the fixed name parameterinfo.xml.

- File attachment for collected parameters

Any file can be used for parameter information. This file is handled as an attachment.

Discovery script format (batch files) [Windows]

Create the discovery script for batch files (discover.cmd) using the format below.

The -dir option must be specified as the first argument (%1), and the directory to which the parameter information should be output must be specified as the second argument (%2). Information about the parameters collected in this directory is output to the parameter information XML file (parameterinfo.xml) and to the file attachment. The parameter information XML file must be output. However, depending on the specifications of the parameter collection definition, the file attachment might not always be output correctly. The script returns 0 if successful, or another value otherwise. The standard output and standard error output are directed to the agent log.

```
@echo off
setlocal
@rem Process to collect parameters from software
<Processing for each software product>
<Process to output parameter information to directory specified as %2 >
@rem Return results (normal)
if ERRORLEVEL 1 goto ERROR_END
endlocal
exit /B 0
@rem Return results (error)
: ERROR_END
```

```
echo ERROR0002 Parameter collecting failed. 1>&2
endlocal
exit /B 1
```

Discovery script format (shell scripts) [Linux]

Create the discovery script for shell scripts (discover.sh) using the format shown below.

The -dir option must be specified as the first argument (\$1), and the directory to which the parameter information should be output must be specified as the second argument (\$2). Information about the parameters collected in this directory is output to the parameter information XML file (parameterinfo.xml) and to the file attachment. Output is not required when there is no file attachment. The script returns 0 if successful, or another value otherwise. The standard output and standard error output are directed to the agent log.

```
#!/bin/sh
# Process to collect parameters from software
<Processing for each software product>
<Process to output parameter information to directory specified as $2 >
# Return results
if [ $? = "0" ]; then
# Returns normal
exit 0
else
# Returns an error
echo "ERROR0002 Parameter collecting failed." 1>&2
exit 1
fi
```

Output directory for discovery script

When using the discovery script to output parameter information, output using the directory format shown below. A parameter information XML file must be created. Only create a file attachment if required.

```
+ <Output directory>
+ Collected parameter information XML file (parameterinfo.xml)
+ File attachment for collected parameters
```

Parameter information XML file format

Create the discovery script so that the parameters collected by this script are output to the parameter information XML file. In this parameter information XML file, specify values for the tags of the keys and values in the parameters. For other tags, specify any value, or omit altogether.

- Tag for which value is specified
parameters
- Tag to be omitted
description
- Tag for which any value is specified
name

An example of the parameter information XML file output is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<parameterInfo version="3.0">
  <name><parm info name></name>
  <parameters>
    <parameter>
      <key><parm key></key>
      <value><parm val></value>
    </parameter>
    ...
  </parameters>
```

```
</parameterInfo>
```

For examples of parameter information XML files, refer to "[4.1.4 Parameter Settings Scripts](#)".

Note

- **Commands that cannot be used with a script**

Do not execute the following commands from a script, because this will cause the script to enter standby status on the business server, and its processing will not complete:

- Commands that require interaction [Windows/Linux]
- Commands for which a window opens during execution [Windows]
- AT commands [Windows]
- Shell scripts created using PowerShell [Windows]
- Commands running in full-screen mode [Linux]

- **Parameter collection scripts**

- Order of execution

The parameter collection scripts do not operate in any particular order.

- **Discovery scripts**

- Execution privileges

Must be executed by an Administrator in Windows.

Must be executed by a superuser in Linux.

- Current directory

The current directory is the path in which the discovery script files are stored.

- Execution privileges [Linux]

Configured automatically during execution.

- Line feed

In Windows, use CR+LF.

In Linux, use LF.

- Byte order mark (BOM) [Linux]

Do not include the UTF-8 byte order mark (BOM) in shell scripts.

- **File attachments**

- Execution privileges [Linux]

If a file attachment includes a shell script, configure execution privileges within the discovery script of the shell script.

4.2.3 Package Files

A package file refers to one or more compressed files in ZIP format. These files are used to associate the parameter collection script with the parameter collection definition. Package files are decompressed on the server once they have been forwarded. They are also used to store file attachments, which are output by the discovery script of the parameter collection script, on the admin server.

The package file to be associated with the parameter collection definition is known as the script package. The compressed file in ZIP format containing the file attachment output by the discovery script of the parameter collection script is known as the parameter package. The parameter collection scripts contained in each of these package files is different. An explanation is provided below:

- Script package

Includes the discovery script of the parameter collection script, and the file attachment. The discovery script must be included.

- Parameter package

Includes the file attachment output by the discovery script of the parameter collection script. The parameter information XML file (parameterinfo.xml) is not included. Note that no parameter package will be created if there is no file attachment.

Script package

A script package is a ZIP file that contains a discovery script and a file attachment. This ZIP file must be stored using the structure shown below. Always store the discovery script in the ZIP file root. The file attachment can contain a directory.

```
<ZIP file root>
+ <Discovery script (discover.cmd or discover.sh)>
+ <File attachment>
```

Parameter package

A parameter package is a ZIP file that contains a file attachment. This ZIP file must be stored using the structure shown below. It also contains a directory if the file attachment contains a directory.

```
<ZIP file root>
+ <File attachment>
```

Execution example of the parameter collection script

The script package is forwarded to the server and decompressed on it. The parameter collection script is stored in the work directory on the server as described below - note that the directory in which the discovery script is stored becomes the current directory, and the discovery script is executed.

```
<work directory>
+ <software id>
+ Discovery script (discover.cmd or discover.sh)
+ Script package file attachment
```

A description of each directory is provided below:

- Software ID directory

Directory unique to each software product, with the name <software id>.

Stores the discovery script and the file attachment included in the script package.

The discovery script outputs the parameter information to the directory below (the file attachment for the collected parameters is included in the parameter package).

```
+ <output directory>
+ Collected parameter information XML file (parameterinfo.xml)
+ File attachment for collected parameters
```



Note

- Package files

- Store the discovery script in the ZIP file root, not in the directory, otherwise it cannot be executed.

- Package files forwarded to the server are deleted once the script has been executed.

4.2.4 Association with the Software

Although the parameter collection definition defines the parameters that can be collected from the software, it does not specify the corresponding software. For this reason, the parameter collection definition must be associated with the software information.

Multiple software information items can be associated for each parameter collection definition. For this reason, parameter collection definitions can be aggregated as a single item, even if there are multiple editions of the software. However, since Windows and Linux editions cannot coexist, distinct parameter collection definitions must be created in such cases.

Moreover, multiple parameter collection definitions can be associated with a single software information item. However, you cannot specify a duplicate key in the parameter list for these parameter collection definitions. Even if you associate multiple parameter collection definitions, one parameter list is collected for one software product.

Chapter 5 Command Reference

5.1 Command List

The following table lists commands.

Type	Function	Command	Overview
5.2 Information Collection Source Managemen t Commands	Information Collection Source Management Command	swcfmg_source_def	Registers, updates, deletes, displays details, lists, uploads, and downloads information collection sources managed by Systemwalker Software Configuration Manager.
5.3 Software Managemen t Commands	Software information management command	swcfmg_software	Registers, updates, deletes, and lists software information managed by Systemwalker Software Configuration Manager.
	Installed software information management command	swcfmg_installedsoftware	Registers, deletes, and lists software information installed on the managed server.
5.4 Patch Managemen t Commandss	Patch Information Management Command	swcfmg_patch_def	Registers, updates, deletes, and lists patch information managed by Systemwalker Software Configuration Manager.
	Applied Patch Information Management Command	swcfmg_patch_applied	Imports and exports patch information applied to the managed server.
5.5 Parameter Managemen t Commands	Parameter settings definition management command	swcfmg_param_settingdef	Adds, updates, deletes, lists, and acquires parameter settings definitions. Also, uploads or downloads the script package associated with parameter settings definitions.
	Parameter collection definition management command	swcfmg_param_collectingdef	Adds, updates, deletes, lists, and acquires parameter collection definitions. Also, uploads or downloads the script package associated with the script package of a parameter collection definition.
	Command to associate software and parameter definitions	swcfmg_param_defassoc	Associates the software and parameter settings definitions, or the software and parameter collection definition. The command can also be used to cancel such associations, as well as display a list of associations.



Note

- Some commands provided by Systemwalker Software Configuration Manager cannot be executed simultaneously. An exclusive control error will occur if a command is executed while another command is still executing.

- In order to execute the commands provided by Systemwalker Software Configuration Manager, the environment variables that are set up during installation must be enabled.

For this reason, execute commands from a command prompt that is opened after Systemwalker Software Configuration Manager has been installed.

5.2 Information Collection Source Management Commands

This section explains the information collection source commands.

5.2.1 swcfmg_source_def (Information Collection Source Management Command)

Description

Registers, updates, performs batch updates, deletes, lists, displays details, uploads, and downloads the information collection sources managed by Systemwalker Software Configuration Manager.

Synopsis

[Windows]

```
swcfmg_source_def.exe
{ -add -id "Information collection source ID" -softitem "Software collection item information file path" [-patchitem "Patch collection item information file path"] -file "Script package file path" [-e "Character encoding"] |
-update -id "Information collection source ID" [-softitem "Software collection item information file path"] [-patchitem "Patch collection item information file path"] [-e "Character encoding"] |
-delete -id "Information collection source ID" |
-list [-file Output file path] [-e "Character encoding"] }
-get -id "Information collection source ID" [-softitem "Collection item information file path"] [-patchitem "Collection item information file path"] [-e "Character encoding"] |
-upload -id "Information collection source ID" -file "Script package file path" |
-download -id "Information collection source ID" -dir "Output directory path" }
```

[Linux]

```
swcfmg_source_def
{ -add -id "Information collection source ID" -softitem "Software collection item information file path" [-patchitem "Patch collection item information file path"] -file "Script package file path" [-e "Character encoding"] |
-update -id "Information collection source ID" [-softitem "Software collection item information file path"] [-patchitem "Patch collection item information file path"] [-e "Character encoding"] |
-batchupdate -file "Information collection source file path" [-e "Character encoding"] |
-delete -id "Information collection source ID" |
-list [-file Output file path] [-e "Character encoding"] }
-get -id "Information collection source ID" [-softitem "Collection item information file path"] [-patchitem "Collection item information file path"] [-e "Character encoding"] |
-upload -id "Information collection source ID" -file "Script package file path" |
-download -id "Information collection source ID" -dir "Output directory path" }
```

Options

-add -id "Information collection source ID" -softitem "Software collection item information file path" [-patchitem "Patch collection item information file path"] -file "Script package file path"

Registers the information collection source specified using the information collection source ID. Specify the collection items of software and patches for the -softitem and -patchitem option. Specify the collection items using a CSV file. For details on the file, refer to "[6.2.2 Collection Item Information File](#)". Specify a script package for the -file option. For details on script packages, refer to "[1.1.1.2 Software Patch Collection Script](#)".

-update -id "Information collection source ID" [-softitem "Software collection item information file path"] [-patchitem "Patch collection item information file path"]

Updates the collection item of the information collection source specified using the information collection source ID. Specify the collection items of software and patches to be updated for the -softitem and -patchitem options. Specify the collection items using a CSV file. For details on the file, refer to ["6.2.2 Collection Item Information File"](#).

-batchupdate -file "Information collection source file path"

Based on the details described in the specified information collection source definition file, performs batch update of multiple information collection sources. The information collection source definition file is in CVS format. For details on the information collection source definition file, refer to ["6.2.1 Information Collection Source File"](#). Collection items cannot be updated using this option. Perform updates of collection item using the -update option of this command.

-delete -id "Information collection source ID"

Deletes the information collection source specified using the information collection source ID.

-list [-file "Output file path"]

Displays a list of the registered information collection sources. The information is output in CSV format. If -file "Output file path" is omitted, information will be output to the standard output for the command. If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten. If output to the file fails due to an error, no new file will be created. The existing file cannot be overwritten. For details on the output file, refer to ["6.2.1 Information Collection Source File"](#).

-get -id "Information collection source ID" [-softitem "Collection item definition file path"] [-patchitem "Collection item definition file path"]

Outputs the details of the information collection source specified using the information collection source ID. Outputs the collection items of software and patches to the file path specified for the -softitem and -patchitem options. The file is in CSV format. For details on the file, refer to ["6.2.2 Collection Item Information File"](#). If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten. If output to the file fails due to an error, no new file will be created. The existing file cannot be overwritten.

-upload -id "Information collection source ID" -file "Script package file path"

Uploads the script package of the information collection source specified using the information collection source ID. Upload a script package to update it. The script package is a zip file. When the following conditions are satisfied, the script package can be uploaded. When the following conditions are not satisfied, an error occurs.

- File extension must be .zip.

 - Ensure that the file extension is .zip.

- File size must be no greater than 2 MB.

 - The maximum uploadable file size is 2 MB.

- Store the directories for OS types in the root.

 - Store the OS type directories (Windows, Linux, or Solaris) in the ZIP file root.

-download -id "Information collection source ID" -dir "Output directory path"

Downloads the script package of the information collection source specified using the information collection source ID. Outputs script package ZIP files to the output directory path.

Ensure that no files exist in the output directory, or an error will occur. When a file exists in the output directory, an error occurs.

-e "Character encoding"

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return values

Return Value	Meaning
0	Completed normally.
10	A parameter error has occurred.
20	There is an error in the specified I/O file path.
30	The specified character encoding is incorrect.
50	The file description is incorrect.
70	There is an error in the specified information collection source ID.
100	Systemwalker Software Configuration Manager is not running.
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

```
<Systemwalker Software Configuration Manager installation directory>\SWCFMGM\bin
```

[Linux]

```
/opt/FJSVcfmgm/bin
```

Privilege required for execution

[Windows]

- Administrator privileges are required. Execute this command as an administrator.
- This command can be executed on the admin server.

[Linux]

- This command can only be executed by the system administrator (superuser).
- This command can be executed on the admin server.

Notes

Common notes

- Commands provided by Systemwalker Software Configuration Manager cannot be executed simultaneously. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.

Notes regarding registration

- For the information collection source ID, specify a character string beginning with an alphanumeric character and containing up to 16 alphanumeric characters, hyphens (-), and underscores (_). The following IDs cannot be used.
 - Direct
 - UpdateAdvisor
 - ROR
 - OpenStack

- When describing the collection item information file, note the following:
 - Describe the collection item for each software and patch separately.
 - Describe one collection item per line. In addition, the same collection item cannot be described multiple times.
 - For details on available collection items, refer to "[1.1.2 List of Collection Items](#)".
 - When defining the software collection item, note the following:
 - Be sure to include a software name and an OS type in the identification item.
 - The installation location cannot be used as an identification item.
 - When defining the patch collection item, note the following:
 - Be sure to include a patch name and an OS type in the identification item.
 - Be sure to define the target software as a definition item. It cannot be used as an identification item.
- When registering an information collection source with the same ID as a deleted information collection source, perform discovery of the software information once before registration.
- Be sure to define the software collection item. When the patch information cannot be collected from the information collection source, it is not necessary to define the patch collection item.
- When there is an OS type which cannot be collected, it is not necessary to register the software patch collection script for the OS type.
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding specified in the collection item information file. Ensure that the character encoding matches.

Notes regarding updating

- Be sure to specify the collection item of either the software or patches.
- The information collection source cannot update the collection item of UpdateAdvisor.
- When describing the collection item information file, note the following:
 - Describe the collection item for each software and patch separately.
 - Describe one collection item per line. In addition, the same collection item cannot be described multiple times.
 - Describe all collection items to be defined including any items which are not to be updated.
 - For details on available collection items, refer to "[1.1.2 List of Collection Items](#)".
 - When defining the collection item of the installation location for the software collection item, the collection item of the installation location cannot be deleted later.
 - Identification items cannot be added or deleted.
 - The target software must be defined in the patch collection items as a definition item.
 - Depending on the updated collection item, perform the necessary revisions of the software patch collection script, and upload the software patch collection script.
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding specified in the collection item information file. Ensure that the character encoding matches.

Notes regarding batch updating

- Using this option, registration and deletion of information collection sources cannot be performed. To perform these operations, use the -add and -delete options.
- Manually registered information collection sources cannot be enabled.

Notes regarding deletion

- The information collected from a deleted information collection source is deleted when discovery of software information is performed after the deletion of the information collection source using this option.
- Manually registered information collection sources and the UpdateAdvisor information collection source cannot be deleted.

Notes regarding displaying details

- When the patch collection item is not defined in the specified information collection source, the patch collection item is not output.

Notes regarding uploading and downloading

- The information collection source cannot upload and download UpdateAdvisor and manually registered script packages.
- When downloading, the file name of the script package is output in the following format:

```
Information collection source ID + ".zip"
```

Notes regarding registration and update errors

- If there is an error in the collection item information file, registration and update of the collection item will be canceled. An error message similar to the following will be output to the trace log:

```
If there is an error in the collection item information file, registration and update of the collection item will be canceled. An error message similar to the following will be output to the trace log:
```

One of the following messages will be output in [Cause]:

Message	Meaning
Number of elements.	Review the number of elements in the line indicated.
The specified elements are invalid.	An invalid value is entered in the element in "(" in the line indicated. Enter a valid value. For details on available values, refer to "6.2.2 Collection Item Information File".
Mandatory elements are not specified.	Enter the mandatory items in the line indicated.
The same collection item is specified in another line.	The same collection item is already specified in one of the preceding lines.
Mandatory identification items are not defined.	Required identification items have not been defined.
Mandatory collection items are not defined.	Required collection items have not been defined.

Examples

- Registration

[Windows]

```
swcfmg_source_def.exe -add -id SourceDefA -softitem C:\work\softwareitem.csv -patchitem C:\work\patchitem.csv
```

[Linux]

```
swcfmg_source_def -add -id SourceDefA -softitem /tmp/softwareitem.csv -patchitem /tmp/patchitem.csv
```

- Update

[Windows]

```
swcfmg_source_def.exe -update -id SourceDefA -softitem C:\work\softwareitem.csv -patchitem C:\work\patchitem.csv
```

[Linux]

```
swcfmg_source_def -update -id SourceDefA -softitem /tmp/softwareitem.csv -patchitem /tmp/patchitem.csv
```

- Batch update

[Windows]

```
swcfmg_source_def.exe -batchupdate -file C:\work\soucedef.csv
```

[Linux]

```
swcfmg_source_def -batchupdate -file /tmp/soucedef.csv
```

- Deletion

[Windows]

```
swcfmg_source_def.exe -delete -id SourceDefA
```

[Linux]

```
swcfmg_source_def -delete -id SourceDefA
```

- List display

[Windows]

```
swcfmg_source_def.exe -add -file C:\work\soucedef.csv
```

[Linux]

```
swcfmg_source_def -add -file /tmp/soucedef.csv
```

- Detail display

[Windows]

```
swcfmg_source_def.exe -get -id SourceDefA -softitem C:\work\softwareitem.csv -patchitem C:\work\patchitem.csv
```

[Linux]

```
swcfmg_source_def -get -id SourceDefA -softitem C:\work\softwareitem.csv -patchitem C:\work\patchitem.csv
```

- Upload

[Windows]

```
swcfmg_source_def.exe -upload -id SourceDefA -file C:\work\sourcedef.zip
```

[Linux]

```
swcfmg_source_def -upload -id SourceDefA -file /tmp/sourcedef.zip
```

- Download

[Windows]

```
swcfmg_source_def.exe -download -id SourceDefA -dir C:\work
```

[Linux]

```
swcfmg_source_def.exe -download -id SourceDefA -dir /tmp
```

5.3 Software Management Commands

This section explains the software management commands.

5.3.1 swcfmg_software (Software Information Management Command)

Description

Registers, updates, deletes, and lists software information managed by Systemwalker Software Configuration Manager.

Synopsis

[Windows]

```
swcfmg_software.exe

{ -add [-sourceid "Information collection source ID"] -file "Software information file path" [-e "Character encoding"] |
-update [-sourceid "Information collection source ID"] -file "Software information file path" [-e "Character encoding"] |
-delete -id "Software ID",... |
-list [-sourceid "Information collection source ID"] [-file "Output file path"] [-e "Character encoding"] }
```

[Linux]

```
swcfmg_software

{ -add [-sourceid "Information collection source ID"] -file "Software information file path" [-e "Character encoding"] |
-update [-sourceid "Information collection source ID"] -file "Software information file path" [-e "Character encoding"] |
-delete -id "Software ID",... |
-list [-sourceid "Information collection source ID"] [-file "Output file path"] [-e "Character encoding"] }
```

Options

-add [-sourceid "*Information collection source ID*"] -file "*Software information file path*"

Registers software information using the specified software information file. The software information is registered as software of the information collection source specified using the information collection source ID. When the information collection source ID is omitted, it is registered as manually registered software.

The software information file is in CSV format. Refer to "[6.3.1 Software Information File](#)" for information on software information files.

-update [-sourceid "*Information collection source ID*"] -file "*Software information file path*"

Updates software information using the specified software information file. Updates the software of the information collection source specified using the information collection source ID. When omitting the information collection source ID, update the manually registered software.

The software information file is in CSV format. Refer to "[6.3.1 Software Information File](#)" for information on software information files.

-delete -id <*Software ID*>,...

Deletes registered software information. Multiple software IDs can be specified by delimiting them with ',' (comma).

The software ID is configured during registration of this command or the discovery of software application. Specifying **-list** allows the software ID to be checked from the software information output.

-list [-sourceid "*Information collection source ID*"] [-file "*Output file path*"]

Outputs registered software information in CSV format. Outputs only the software of the information collection source specified using the information collection source ID. When the information collection source ID is omitted, the manually registered software is output.

If **-file <*Output file path*>** is omitted, information will be output to the standard output for the command.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If the file cannot be output because of an error, either a new file will not be created or, if an existing file was to be overwritten, the existing file will be kept as it is, without being overwritten.

-e <Character encoding>

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return value

Return value	Meaning
0	Completed normally.
10	A parameter error has occurred.
20	There is an error in the specified I/O file path.
30	The specified character encoding is incorrect.
40	Failed to read the software information file.
50	The software information file description is incorrect.
70	The specified information collection source ID or software ID does not exist.
80	The software information targeted for deletion cannot be deleted because the parameter settings information or the installed software information has been registered.
100	Systemwalker Software Configuration Manager is not running.
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

`<Systemwalker Software Configuration Manager installation directory>\SWCFMGM\bin`

[Linux]

`/opt/FJSVcfmgm/bin`

Privilege required for execution

[Windows]

- Administrator privileges are required. Execute this command as an administrator.
- This command can be executed on the admin server.

[Linux]

- This command can only be executed by the system administrator (superuser).

- This command can be executed on the admin server.

Notes

Common notes

- Commands provided by Systemwalker Software Configuration Manager cannot be executed simultaneously. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.
- Using this command, the information collection source cannot register, update, or delete the software of UpdateAdvisor.
- Only the value of the definition item can be registered and updated.

Notes regarding registration

- Software information is registered in accordance with the software information file.
- The value of the identification item cannot be omitted. When all identification items are the same, software are recognized as being the same. Furthermore, when the same software is already registered, an error occurs.
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding used in the software information file. Ensure that the character encoding matches.

Notes regarding updating

- When updating registered software information, it is advisable to edit the file that is listed, then treat the edited file as the input file for updating.
- The value of the identification item cannot be omitted. In addition, the value of the identification item cannot be updated.
- An error will occur in the input file if the description does not include *<Software ID>*. When updating, always include *<Software ID>*.

Notes regarding deletion

- Use this command to delete software information that no longer needs to be managed by Systemwalker Software Configuration Manager.
- Before deleting software information, confirm the following:
 - The corresponding software has not been registered as installed software.
 - Patches for the corresponding software are not registered

Notes regarding list display

- If no software information has been registered, only the header information will be output.

```
#software-id,software-info,...
```

Notes regarding registration and update errors

- If there is an error in the software information file, registration and update of the software information will be canceled. An error message similar to the following will be output to the standard output:

```
The file description is incorrect.[Line number:<n>](Cause:<Cause>)
```

One of the following messages will be output in *<Cause>*:

Message	Meaning
Number of elements.	Review the number of elements in the line indicated.
Mandatory elements are not specified.	Enter the mandatory items in the line indicated.
The same software is specified in another line.	The same software information is already specified in one of the preceding lines.

Message	Meaning
The specified elements are invalid.	An invalid value is entered in the element in "(" in the line indicated. Enter a valid value. For details on available values, refer to " 6.3.1 Software Information File ". Omit the values of collection items which are not definition items.
The specified software does not exist.	Software information corresponding to "software-id" in the line indicated does not exist. Specify the existing software ID.
The software has already been registered.	The software information in the line indicated has already been registered.

Examples

- Registration

[Windows]

```
swcfmg_software.exe -add -file C:\work\softwareinfo.csv
```

[Linux]

```
swcfmg_software -add -file /tmp/softwareinfo.csv
```

- Update

[Windows]

```
swcfmg_software.exe -update -file C:\work\softwareinfo.csv
```

[Linux]

```
swcfmg_software -update -file /tmp/softwareinfo.csv
```

- Deletion

[Windows]

```
swcfmg_software.exe -delete -id 34GR4GJLJ@Direct,GIDFGJO43@Direct,5YUYJG935@Direct
```

[Linux]

```
swcfmg_software -delete -id 34GR4GJLJ@Direct,GIDFGJO43@Direct,5YUYJG935@Direct
```

- List display

[Windows]

```
swcfmg_software.exe -list -file C:\work\softwareinfo.csv
```

[Linux]

```
swcfmg_software -list -file /tmp/softwareinfo.csv
```

Execution results/output format

- List display

Standard output

```
#software-id,software-info,...
34GR4GJLJ@Direct,ABCDE-Software,1.0,windows,ABCDE Company file output
```

```
Output of the software information has started.  
Output of the software information has ended.
```

Note that the following header information is added to the first line of the CSV file that is output:

```
#software-id,software-info,...
```

5.3.2 swcfmg_installedsoftware (Installed Software Information Management Command)

Description

Registers, deletes, and lists software information installed on the managed server.

Use this command to register installed software information so that the information collection source will display the manually registered software in the management console as installed software and configure the software parameters.

Synopsis

[Windows]

```
swcfmg_installedsoftware.exe  
{ -add -file <Installed software information file path> [-e <Character encoding>] |  
-delete -file <Installed software information file path> [-e <Character encoding>] |  
-list [{-serverid <Server ID>,... | -ip <IP address>,...} ] [-file <Output file path>] [-e<Character encoding>] [-all] }
```

[Linux]

```
swcfmg_installedsoftware  
{ -add -file <Installed software information file path> [-e <Character encoding>] |  
-delete -file <Installed software information file path> [-e <Character encoding>] |  
-list [{-serverid <Server ID>,... | -ip <IP address>,...} ] [-file <Output file path>] [-e<Character encoding>] [-all] }
```

Options

-add -file <Installed software information file path>

Registers installed software information using the specified installed software information file

The installed software information file is in CSV format. Refer to "[6.3.2 Installed Software Information File](#)" for information on installed software information files.

-delete -file <Installed software information file path>

Deletes installed software information using the specified installed software information file.

The installed software information file is in CSV format. Refer to "[6.3.2 Installed Software Information File](#)" for information on installed software information files.

-list [{-serverid <Server ID>,... | -ip <IP address>,...}] [-file <Output file path>] [-all]

Use this command to output registered installed software information in CSV format.

-serverid <Server ID>,... | -ip <IP address>,...

Specifies the server ID or the IP address of the server that outputs the installed software information.

If this option is omitted, the installed software information on all servers will be output.

[-file <Output file path>]

Specify the output file path.

If this option is omitted, information will be output to the standard output for the command.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

[-all]

The information collection source outputs all the installed software information included in all of the software of the information collection source in addition to the manually registered installed software.

-e <Character encoding>

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return value

Return value	Meaning
0	Completed normally.
10	A parameter error has occurred.
20	There is an error in the specified I/O file path.
30	The specified character encoding is incorrect.
40	Failed to read the installed software information file.
50	The installed software information file description is incorrect.
70	The specified server does not exist.
100	Systemwalker Software Configuration Manager is not running.
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

<Systemwalker Software Configuration Manager installation directory>\SWCFMGM\bin

[Linux]

/opt/FJSVcfmgm/bin

Privilege required for execution

[Windows]

- Administrator privileges are required. Execute this command as an administrator.
- This command can be executed on the admin server.

[Linux]

- This command can only be executed by the system administrator (superuser).
- This command can be executed on the admin server.

Notes

Common notes

- Commands provided by Systemwalker Software Configuration Manager cannot be executed simultaneously. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.

Notes regarding registration

- An error will occur if the server with the server ID or the IP address listed in the installed software information file is not registered in Systemwalker Software Configuration Manager as a managed server.
- An error will occur if software information with the software ID listed in the installed software information file is not registered in Systemwalker Software Configuration Manager.
- When registering a multi-instance installed software, specify the software IDs and installation locations separated by semicolons (";") for software-id.
- When the information collection source registers the information of installed software which was not manually registered during the discovery of software patch information, the corresponding installed software information is deleted when the information registered using this command cannot be collected.
- An error will occur if a combination of registered server ID and software ID or IP address and software ID already exists in the installed software information file.
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding used in the installed software information file. Ensure that the character encoding matches.

Notes regarding deletion

- Use this command to delete installed software information if the installed software information was erroneously set for the managed server or the software was uninstalled from the server in question.
- Use this command to delete the installed software information if a managed server registered in Systemwalker Software Configuration Manager is deleted.
- An error will occur if the server with the server ID or the IP address listed in the installed software information file is not registered in Systemwalker Software Configuration Manager as a managed server.
- An error will occur if software information with the software ID listed in the installed software information file is not registered in Systemwalker Software Configuration Manager.
- When deleting a multi-instance installed software, specify the software IDs and installation locations separated by semicolons (";") for software-id.
- When the information collection source deletes the information of installed software which was not manually registered during the discovery of software patch information, the corresponding installed software information is reregistered when the information deleted using this command can be collected.
- An error will occur if, in the combination of server ID and software ID or IP address and software ID specified in the installed software information file, the relevant server exists but the software ID has not been set as installed software.
- Installed software to which patches have already been applied cannot be deleted.
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding used in the installed software information file. Ensure that the character encoding matches.

Notes regarding list display

- If no installed software information has been registered, only the header information will be output.

```
#host-name,servergroup-name,server-id,ipaddress,software-id,software-name,version,os-type,vendor
```

Notes regarding registration and deletion errors

- If there is an error in the installed software information file, registration and deletion of the installed software information will be canceled. An error message similar to the following will be output to the trace log:

```
The file description is incorrect. [Line number:n] (Cause:<Cause>)
```

One of the following messages will be output in <Cause>:

Message	Meaning
Number of elements.	Review the number of elements in the line indicated.
The specified elements are invalid.(software-id)	<ul style="list-style-type: none">- The installation location has not been defined for the collection item of the software in the line indicated.- The OS type of the software in the line indicated is not the OS type of the server.
Mandatory elements are not specified.	Enter the mandatory items in the line indicated.
The same server and software id are specified in another line.	A combination of the same server and software ID is already specified in one of the preceding lines.
The specified server does not exist.	The server corresponding to "server" in the line indicated does not exist. Specify the server ID or IP address of the existing server.
The specified software does not exist.	Software information corresponding to "software-id" in the line indicated does not exist. Specify the existing software ID.
The installed software has already been registered.	The software information in the line indicated has already been registered. If you re-register a managed server, the installed software information that was registered previously may still exist. In this case, use the -list option to check the installed software information and delete the relevant installed software information before re-registering it.
The specified installed software does not exist.	The combination of server ID and software ID or IP address and software ID in the line indicated has not been registered as installed software information.
The specified installed software cannot be deleted.	As patches have been applied to the installed software in the line indicated, the software cannot be deleted.
Installation directory is not specified.	The specified software is installed as a multi-instance software. Specify the installation location.

Examples

- Registration

[Windows]

```
swcfmg_installedsoftware.exe -add -file C:\work\installedsoftwareinfo.csv
```

[Linux]

```
swcfmg_installedsoftware -add -file /tmp/installedsoftwareinfo.csv
```

- Deletion

[Windows]

```
swcfmg_installedsoftware.exe -delete -file C:\work\installedsoftwareinfo.csv
```


[Linux]

```
swcfmg_installedsoftware -delete -file /tmp/installedsoftwareinfo.csv
```

- List display

[Windows]

```
swcfmg_installedsoftware.exe -list -file C:\work\installedsoftwareinfo.csv
```

[Linux]

```
swcfmg_installedsoftware -list -file /tmp/installedsoftwareinfo.csv
```

Execution results/output format

- List display

Standard output

```
#host-name,servergroup-name,server-id,ipaddress,software-id,software-name,version,os-  
type,vendor  
Server01,Servergroup01,UB5IFK3AT@Direct,10.10.10.11,UDP000001,ABCDE-Software,1.0,Windows,ABCDE  
Company  
Server01,Servergroup01,UB5IFK3AT@Direct,10.10.10.11,UDP000002,FGHIJ-Software,1.0,Windows,ABCDE  
Company  
Server02,Servergroup02,FGGF436GF@Direct,10.10.10.12,UDP000001,ABCDE-Software,1.0,Windows,ABCDE  
Company
```

File output

```
Output of the software information has started.  
Output of the software information has ended.
```

Note that the following header information is added to the first line of the CSV file that is output:

```
#host-name,servergroup-name,server-id,ipaddress,software-id,software-name,version,os-  
type,vendor
```

5.4 Patch Management Commands

This section explains the patch management commands.

5.4.1 swcfmg_patch_def (Patch Information Management Command)

Description

Registers, updates, deletes, and lists patch information managed by Systemwalker Software Configuration Manager.

Synopsis

[Windows]

```
swcfmg_patch_def.exe  
{ -add [-sourceid "Information collection source ID"] -file "Patch information file path" [-e "Character encoding"] |  
-update [-sourceid "Information collection source ID"] -file "Patch information file path" [-e "Character encoding"] |  
-delete -id "Patch ID",... |  
-list [-sourceid "Information collection source ID"] [-file "Output file path"] [-e "Character encoding"] }
```

[Linux]

```
swcfmg_patch_def
```

```
{ -add [-sourceid "Information collection source ID"] -file "Patch information file path" [-e "Character encoding"] |  
-update [-sourceid "Information collection source ID"] -file "Patch information file path" [-e "Character encoding"] |  
-delete -id "Patch ID",... |  
-list [-sourceid "Information collection source ID"] [-file "Output file path"] [-e "Character encoding"] }
```

Options

-add [-sourceid "*Information collection source ID*"] -file "*Patch information file path*"

Registers patch information using the specified patch information file. The patch information is registered as a patch of the information collection source specified using the information collection source ID. When the information collection source ID is omitted, it is registered as a manually registered patch.

The patch information file is in CSV format. For details on the patch information file, refer to "[6.4.1 Patch Information File](#)".

-update [-sourceid "*Information collection source ID*"] -file "*Patch information file path*"

Updates patch information using the specified patch information file. Updates the patches of the information collection source specified using the information collection source ID. When the information collection source ID is omitted, a manually registered patch is updated.

The patch information file is in CSV format. For details on the patch information file, refer to "[6.4.1 Patch Information File](#)".

-delete -id "*Patch ID*",...

Specify a patch ID to delete registered patch information. Multiple patch IDs can be specified by delimiting them with "," (commas).

The patch IDs are configured during registration of this command or discovery of the software information. Specifying **-list** enables patch information to be checked from the output patch information.

-list [-sourceid "*Information collection source ID*"] [-file "*Output file path*"]

Outputs registered patch information in CSV format.

If **-file** "*Output file path*" is omitted, information will be output to the standard output for the command.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

For details on the output file, refer to "[6.4.1 Patch Information File](#)".

-e "*Character encoding*"

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return values

Return Value	Meaning
0	Completed normally.
10	A parameter error has occurred.
20	There is an error in the specified I/O file path.
30	The specified character encoding is incorrect.
40	Failed to read the patch information file.
50	The patch information file description is incorrect.

Return Value	Meaning
70	The specified information collection source ID or patch ID does not exist.
80	As the applied patch information is registered in the patch information to be deleted, it cannot be deleted.
100	Systemwalker Software Configuration Manager is not running.
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

```
Systemwalker Software Configuration Manager installation directory\SWCFMGM\bin
```

[Linux]

```
/opt/FJSVcfmgm/bin
```

Privilege required for execution

[Windows]

- The command can be executed on the admin server.

[Linux]

- The command can be executed on the admin server.

Notes

Common notes

- Commands provided by Systemwalker Software Configuration Manager cannot be executed simultaneously. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.
- Using this command, the information collection source cannot register, update, or delete the patches of UpdateAdvisor.
- Only the value of the definition item can be registered and updated.

Notes regarding registration

- Patch information is registered in accordance with the patch information file.
- The value of the identification item cannot be omitted. When all identification items are the same, patches are recognized as being the same. Therefore, when the same patch is already registered, an error occurs.
- Note the following when specifying a value for the collection item of a patch.
 - The collection item of the target software cannot be omitted.
 - Specify the software ID of the software to be patched in the collection item of the target software. When there are multiple software to be patched, specify the software separated by semicolons (";").
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding specified in the patch information file. Ensure that the character encoding matches.

Notes regarding updating

- When updating registered patch information, it is advisable to edit the file that is listed, then treat the edited file as the input file for updating.
- The value of the identification item cannot be omitted. In addition, the value of the identification item cannot be updated.
- An error will occur in the input file if the description does not include "Patch ID". When updating, always include "Patch ID".
- Note the following when specifying a value for the collection item of a patch.
 - The collection item of the target software cannot be omitted.
 - Specify the software ID of the software to be patched in the collection item of the target software. When there are multiple software to be patched, specify the software separated by semicolons (";").
 - Do not delete the software to which the corresponding patches have already been applied from the collection items of the target software.

Notes regarding deletion

- Use this command to delete patch information that no longer needs to be managed by Systemwalker Software Configuration Manager.
- Before deleting patch information, confirm the following:
 - The corresponding patch is not registered as an applied patch

Notes regarding list display

- If no patch information has been registered, only the header information will be output.

```
#patch-id,patch-info,...
```

Notes regarding registration and update errors

- If there is an error in the patch information file, registration and update of the patch information will be canceled. An error message similar to the following will be output to the standard output:

```
CFMGC20005 The file description is incorrect. [Line number:n][Cause: Cause]
```

One of the following messages will be output in [Cause]:

Message	Meaning
Number of elements.	Review the number of elements in the line indicated.
Mandatory elements are not specified.	Enter the mandatory items in the line indicated.
The same patch is specified in another line.	The same patch information is already specified in one of the preceding lines.
The specified elements are invalid.	An invalid value is entered in the element in "()" in the line indicated. Enter a valid value. For details on available values, refer to " 6.4.1 Patch Information File ". Omit the values of collection items which are not definition items.
The specified patch does not exist.	Patch information corresponding to "patch-id" in the line indicated does not exist. Specify an existing patch ID.
The patch has already been registered.	The patch information in the line indicated has already been registered.

Examples

- Registration

[Windows]

```
swcfmg_patch_def.exe -add -file C:\work\patchinfo.csv
```

[Linux]

```
swcfmg_patch_def -add -file /tmp/patchinfo.csv
```

- Update

[Windows]

```
swcfmg_patch_def.exe -update -file C:\work\patchinfo.csv
```

[Linux]

```
swcfmg_patch_def -update -file /tmp/patchinfo.csv
```

- Deletion

[Windows]

```
swcfmg_patch_def.exe -delete -id 34GR4GJLJ@Direct,34GR4GJLJ@Direct,34GR4GJLJ@Direc
```

[Linux]

```
swcfmg_patch_def -delete -id 34GR4GJLJ@Direct,34GR4GJLJ@Direct,34GR4GJLJ@Direct
```

- List display

[Windows]

```
swcfmg_patch_def.exe -list -file C:\work\patchinfo.csv
```

[Linux]

```
swcfmg_patch_def -list -file C:\work\patchinfo.csv
```

5.4.2 swcfmg_patch_applied (Applied Patch Information Management Command)

Description

Registers, deletes, and lists patch information applied to a managed server.

The information collection item displays manually registered patches as applied patches in the management console.

Synopsis

[Windows]

```
swcfmg_patch_applied.exe  
{ -add -file "Applied patch information file path" [-e "Character encoding"] |  
-delete -file "Applied patch information file path" [-e "Character encoding"] |  
-list [{-serverid "Server ID",... | -ip "IP address",...} ] [-file "Output file path"] [-e "Character encoding"] [-all] }
```

[Linux]

```
swcfmg_patch_applied  
{ -add -file "Applied patch information file path" [-e "Character encoding"] |  
-delete -file "Applied patch information file path" [-e "Character encoding"] |  
-list [{-serverid "Server ID",... | -ip "IP address",...} ] [-file "Output file path"] [-e "Character encoding"] [-all] }
```

Options

-add -file "*Applied patch information file path*"

Registers applied patch information using the specified applied patch information file.

The applied patch information file is in CSV format. For details on the applied patch information file, refer to ["6.4.2 Applied Patch Information File"](#).

-delete -file "Applied patch information file path"

Deletes applied patch information using the specified applied patch information file.

The applied patch information file is in CSV format. For details on the applied patch information file, refer to ["6.4.2 Applied Patch Information File"](#).

-list [{-serverid "Server ID",... | -ip "IP address",...}] [-file "Output file path"]

Outputs the applied patch information registered using this command in CSV format. This file is in CSV format. For details on the file, refer to ["6.2.3 Collection Information CSV File"](#). If -file "Output file path" is omitted, information will be output to the standard output for the command. If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten. If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

[-file "Output file path"]

Specify the output file path.

If this option is omitted, information will be output to the standard output for the command.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

For details on the output file, refer to ["6.4.2 Applied Patch Information File"](#) in File Reference.

[-all]

Outputs not only the applied patches whose information collection source is manually registered, but all the applied patch information of all information collection sources.

-e "Character encoding"

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return values

Return Value	Meaning
0	Completed normally.
10	A parameter error has occurred.
20	There is an error in the specified I/O file path.
30	The specified character encoding is incorrect.
40	Failed to read the applied patch information file.
50	The applied patch information file description is incorrect.
100	Systemwalker Software Configuration Manager is not running.
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

```
<Systemwalker Software Configuration Manager installation directory>\SWCFMGM\bin
```

[Linux]

```
/opt/FJSVcfmgm/bin
```

Privilege required for execution

[Windows]

- The command can be executed on the admin server.

[Linux]

- The command can be executed on the admin server.

Notes

Common notes

- Commands provided by Systemwalker Software Configuration Manager cannot be executed simultaneously. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.

Notes regarding registration

- An error will occur if the server with the server ID or the IP address listed in the applied patch information file is not registered in Systemwalker Software Configuration Manager as a managed server.
- An error will occur if the software information with the software ID listed in the applied patch information file is not registered in Systemwalker Software Configuration Manager.
- When registering an applied patch for installed software for which the installation destinations are configured, specify the software IDs and installation locations separated by semicolons (";") for software-id.
- When the information collection source registers the information of an applied patch which was not manually registered during the discovery of software patch information, the corresponding applied patch information is deleted when the information registered using this command cannot be collected.
- An error will occur if a combination of a registered server ID or IP address, a software ID, and a patch ID already exists in the applied patch information file.
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding specified in the applied patch information file. Ensure that the character encoding matches.

Notes regarding deletion

- Use this command to delete applied patch information if the applied patch information was wrongly set for the managed server or the patch was deleted from the corresponding server.
- An error will occur if the server with the server ID or the IP address listed in the applied patch information file is not registered in Systemwalker Software Configuration Manager as a managed server.
- An error will occur if the software information with the software ID listed in the applied patch information file is not registered in Systemwalker Software Configuration Manager.
- When deleting applied patch information for installed software for which the installation destinations are configured, specify the software IDs and installation locations separated by semicolons (";") for software-id.

- When the information collection source deletes the information of an applied patch which was not manually registered during the discovery of software patch information, the corresponding applied patch information is reregistered when the information deleted using this command can be collected.
- An error will occur if a combination of a registered server ID or IP address, a software ID, and patch ID is used and the applied patch information is not registered.
- Behavior is not guaranteed if the character encoding specified for this command does not match the character encoding specified in the applied patch information file. Ensure that the character encoding matches.

Notes regarding list display

- If no applied patch information has been registered, only the header information will be output.

```
#server,software-id,patch-id
```

Notes regarding registration and deletion errors

- If there is an error in the applied patch information file, registration and deletion of the applied patch information will be canceled. An error message similar to the following will be output to the trace log:

```
The file description is incorrect. [Line number:n][Cause: Cause]
```

One of the following messages will be output in [Cause]:

Message	Meaning
Number of elements.	Review the number of elements in the line indicated.
The specified elements are invalid.(patch-id)	The patch in the line indicated cannot be applied to the software in the line indicated.
Mandatory elements are not specified.	Enter the mandatory items in the line indicated.
The same server, software id, and patch id are specified in another line.	A combination of the same server, software ID, and patch ID is already specified in one of the preceding lines.
The specified server does not exist.	The server corresponding to "server" in the line indicated does not exist. Specify the server ID or IP address of the existing server.
The specified patch does not exist.	Patch information corresponding to "patch-id" in the line indicated does not exist. Specify an existing patch ID.
The applied patch has already been registered.	The applied patch information in the line indicated has already been registered.
The specified applied patch does not exist.	The combination of the server ID or IP address, software ID, and patch ID in the line indicated has not been registered as applied patch information.

Examples

- Registration

[Windows]

```
swcfmg_patch_applied.exe -add -file C:\work\appliedpatchinfo.csv
```

[Linux]

```
swcfmg_patch_applied -add -file /tmp/appliedpatchinfo.csv
```

- Deletion

[Windows]

```
swcfmg_patch_applied.exe -delete -file C:\work\appliedpatchinfo.csv
```

[Linux]


```
swcfmg_patch_applied -file /tmp/appliedpatchinfo.csv
```

- List display

[Windows]

```
swcfmg_patch_applied.exe -list -file C:\work\appliedpatchinfo.csv
```

[Linux]

```
swcfmg_patch_applied -add -list -file /tmp/appliedpatchinfo.csv
```

5.5 Parameter Management Commands

This section explains the parameter management commands.

5.5.1 swcfmg_param_settingdef (Parameter Settings Definition Management Command)

Description

Adds, updates, deletes, lists, and acquires parameter settings definitions. Also, uploads or downloads the script package associated with parameter settings definitions.

Use a parameter settings definition input file as the input file for adding, updating, or acquiring. The parameter settings definition input file is in XML format. Refer to "[4.1 Definition of Parameters to be Set](#)" for details.

Uses a script package for uploading and downloading. The script package is a zip file. Refer to "[4.1.5 Package Files](#)" for details.

Synopsis

[Windows]

```
swcfmg_param_settingdef.exe  
{ -add -file <Parameter settings definition file path> |  
-update -id <Parameter settings definition ID> -file <Parameter settings definition file path> |  
-delete -id <Parameter settings definition ID>,... |  
-list [-file <Output file path>] [-e <Character encoding>] |  
-get -id <Parameter settings definition ID> -file <Output file path> |  
-upload -id <Parameter settings definition ID> -file <Script package file path> |  
-download -id <Parameter settings definition ID> -dir <Output directory path> }
```

[Linux]

```
swcfmg_param_settingdef  
{ -add -file <Parameter settings definition file path> |  
-update -id <Parameter settings definition ID> -file <Parameter settings definition file path> |  
-delete -id <Parameter settings definition ID>,... |  
-list [-file <Output file path>] [-e <Character encoding>] |  
-get -id <Parameter settings definition ID> -file <Output file path> |  
-upload -id <Parameter settings definition ID> -file <Script package file path> |  
-download -id <Parameter settings definition ID> -dir <Output directory path> }
```

Options

-add -file <Parameter settings definition file path>

Adds the parameter settings definition using the specified parameter settings definition file. The input file is a parameter settings definition XML file.

A parameter settings definition ID is generated for the newly added parameter settings definition. Outputs this ID to standard output.

- Format of ID to be generated

The parameter settings definition ID is automatically assigned a number according to the following rule:

```
SS (prefix) + 8-digit number
Example:
SS00000001
```

- Output format for standard output

The parameter settings definition ID is output in the following format:

```
[<Parameter settings definition ID>]
```

-update -id <Parameter settings definition ID> -file <Parameter settings definition file path>

Updates the parameter settings definition that has the specified ID, using the specified parameter settings definition file path. The input file is a parameter settings definition XML file.

-delete -id <Parameter settings definition ID> ,

Deletes the parameter settings definition from the parameter settings definition ID. Also, deletes the association (if any) between the software and the parameter settings definition. Multiple parameter settings definition IDs can be specified by delimiting them with ',' (comma).

-list [-file <Output file path>]

Uses a parameter settings definition list file in CSV format to output a list of registered parameter settings definitions.

If **-file <Output file path>** is omitted, information will be output to the standard output for the command.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If the file cannot be output because of an error, either a new file will not be created or, if an existing file was to be overwritten, the existing file will be kept as it is, without being overwritten.

- Output format for standard output

Outputs the list of parameter settings definitions in the following format:

Item No.	Item	Remarks
1	Parameter settings definition ID	Outputs the parameter settings definition ID.
2	Is the definition predefined in the product?	Outputs whether the parameter settings definition is predefined in the product.
3	Parameter settings definition name	Outputs the parameter settings definition name.
4	Parameter settings method	Outputs the parameter settings method.
5	Script package file name	Outputs the name of the script package file.

- Header format

Insert the header in the first line of the CSV file using the format shown below. The header is output even if there is no information to be output to the file.

```
#Definition ID,Product predefined,Definition name,Method,Package name
```

-get -id <Parameter settings definition ID> -file <Output file path>

Outputs the parameter settings definition that has the specified ID to the specified output file path. The output file is a parameter settings definition XML file.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

-upload -id <Parameter settings definition ID> -file <Script package file path>

Uploads the script package to the parameter settings definition that has the specified ID. The input file is a script package zip file.

Script packages that can be uploaded are subject to the following conditions. An error will occur if conditions other than these are used.

- File extension must be .zip.

Ensure that the file extension is .zip.

- File size must be no greater than 2 MB.

The maximum uploadable file size is 2 MB.

- Store the startup script in the root.

Store the startup script (startup.cmd or startup.sh) in the root of the zip file rather than under the directory.

-download -id <Parameter settings definition ID> -dir <Output directory path>

Downloads the script package registered in the parameter settings definition that has the specified ID. Outputs script package zip files to the output directory path.

Ensure that no files exist in the output directory, or an error will occur.

-e <Character encoding>

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return value

Return value	Meaning
0	Completed normally.
10	A parameter error has occurred.
20	The specified I/O path is incorrect.
30	The specified character encoding is incorrect.
40	Unable to analyze the file.
50	The file description is incorrect.
70	The specified ID does not exist.
71	The specified ID is product-predefined.
72	The package has not been registered in the specified ID.
73	The package has not been correctly registered in the specified ID.
80	Cannot be deleted because there is a predefined parameter associated with the parameter settings definition targeted for deletion.
100	Systemwalker Software Configuration Manager is not running.

Return value	Meaning
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
120	There is not enough free disk space for the media library.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

```
<Systemwalker Software Configuration Manager installation directory>\SWCFMGM\bin
```

[Linux]

```
/opt/FJSVcfmgm/bin
```

Privilege required/execution environment

[Windows]

- Administrator privileges are required. Execute this command as an administrator.
- This command can be executed on the admin server.

[Linux]

- This command can only be executed by the system administrator (superuser).
- This command can be executed on the admin server.

Notes

Common notes

- This command cannot be executed simultaneously with other commands. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.

Notes regarding update

- The method cannot be changed if the software and a parameter setting definition are associated or a script package is registered.

Notes regarding deletion

- Use this command to delete information about parameter settings definitions that no longer need to be managed by Systemwalker Software Configuration Manager.
- Before deleting a parameter settings definition, ensure that there are no associated predefined parameters. The parameter settings definition cannot be deleted if such information exists.

Notes regarding list display

- If no parameter settings definitions have been registered, only the header information will be output.

```
#Definition ID,Product predefined,Definition name,Method,Package name
```

Notes regarding additions and update errors

- If a parameter settings definition file is invalid, the registration and update of the parameter settings definitions will be canceled and the following error message will be output:

```
An error occurred during XML file validation. [Details:<Details>]
```

Examples

- Add

[Windows]

```
swcfmg_param_settingdef.exe -add -file C:\work\paramsettingdef.xml
```

[Linux]

```
swcfmg_param_settingdef -add -file /tmp/paramsettingdef.xml
```

- Update

[Windows]

```
swcfmg_param_settingdef.exe -update -id SS00000001 -file C:\work\paramsettingdef.xml
```

[Linux]

```
swcfmg_param_settingdef -update -id SS00000001 -file /tmp/paramsettingdef.xml
```

- Delete

[Windows]

```
swcfmg_param_settingdef.exe -delete -id SS00000001
```

[Linux]

```
swcfmg_param_settingdef -delete -id SS00000001
```

- List

[Windows]

```
swcfmg_param_settingdef.exe -list
```

[Linux]

```
swcfmg_param_settingdef -list
```

- Acquire

[Windows]

```
swcfmg_param_settingdef.exe -get -id SS00000001 -file C:\work\paramsettingdef.xml
```

[Linux]

```
swcfmg_param_settingdef -get -id SS00000001 -file /tmp/paramsettingdef.xml
```

- Upload

[Windows]

```
swcfmg_param_settingdef.exe -upload -id SS00000001 -file C:\work\paramsettingdef.zip
```

[Linux]

```
swcfmg_param_settingdef -upload -id SS00000001 -file /tmp/paramsettingdef.zip
```

- Download

[Windows]

```
swcfmg_param_settingdef.exe -download -id SS00000001 -dir C:\work\paramsettingdef
```

[Linux]

```
swcfmg_param_settingdef -download -id SS00000001 -dir /tmp/paramsettingdef
```

5.5.2 swcfmg_param_collectingdef (Parameter Collection Definition Management Command)

Description

Adds, updates, deletes, lists, and acquires parameter collection definitions. Also, uploads or downloads the script package associated with the script package of a parameter collection definition.

Uses a parameter collection definition input file as the input file for adding, updating, and acquiring. The parameter collection definition input file is in XML format. Refer to "[4.2 Definition of Parameters to be Collected](#)" for details.

Uses a script package for uploading and downloading. The script package is a zip file. Refer to "[4.2.3 Package Files](#)" for details.

Synopsis

[Windows]

```
swcfmg_param_collectingdef.exe  
{ -add -file <Parameter collection definition file path> |  
-update -id <Parameter collection definition ID> -file <Parameter collection definition file path>|  
-delete -id <Parameter collection definition ID>,... |  
-list [-file <Output file path>] [-e <Character encoding>] |  
-get -id <Parameter collection definition ID> -file <Output file path>|  
-upload -id <Parameter collection definition ID> -file <Script package file path> |  
-download -id <Parameter collection definition ID> -dir <Output directory path> }
```

[Linux]

```
swcfmg_param_collectingdef  
{ -add -file <Parameter collection definition file path> |  
-update -id <Parameter collection definition ID> -file <Parameter collection definition file path>|  
-delete -id <Parameter collection definition ID>,... |  
-list [-file <Output file path>] [-e <Character encoding>] |  
-get -id <Parameter collection definition ID> -file <Output file path>|  
-upload -id <Parameter collection definition ID> -file <Script package file path> |  
-download -id <Parameter collection definition ID> -dir <Output directory path> }
```

Options

-add -file <Parameter collection definition file path>

Adds the parameter collection definition using the specified parameter collection definition file. The input file is a parameter collection definition XML file.

A parameter collection definition ID is generated for the newly added parameter collection definition. Outputs this ID to standard output.

- Format of ID to be generated

The parameter collection definition ID is automatically assigned a number according to the following rule:

SC (prefix) + 8-digit number Example: SC00000001
--

- Output format for standard output

The parameter collection definition ID is output in the following format:

[<Parameter collection definition ID>]
--

-update -id <Parameter collection definition ID> -file <Parameter collection definition file path>

Updates the parameter collection definition for the parameter collection definition ID using the specified parameter collection definition file. The input file is a parameter collection definition XML file.

-delete -id <Parameter collection definition ID>,...

Deletes the parameter collection definition from the parameter collection definition ID. Also, deletes the association (if any) between the software and the parameter collection definition. Multiple parameter collection definition IDs can be specified by delimiting them with "," (comma).

-list [-file <Output file path>]

Uses a parameter collection definition list file in CSV format to output a list of registered parameter collection definitions.

If -file <Output file path> is omitted, information will be output to the standard output for the command.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

- Output format for standard output

Outputs the list of parameter collection definitions in the following format:

Item No.	Item	Remarks
1	Parameter collection definition ID	Outputs the parameter collection definition ID.
2	Is the parameter collection definition product-predefined?	Outputs whether the parameter collection definition is product-predefined.
3	Parameter collection definition name	Outputs the parameter collection definition name.
4	Parameter collection method	Outputs the parameter collection method.
5	Script package file name	Outputs the name of the script package file.

- Header format

Insert the header in the first line of the CSV file using the format shown below. The header is output even if there is no information to be output to the file.

#Definition ID,Product predefined,Definition name,Method,Package name

-get -id <Parameter collection definition ID> -file <Output file path>

Outputs the parameter collection definition for the parameter collection definition ID to the specified output file path. The output file is a parameter collection definition XML file.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

-upload -id <Parameter collection definition ID> -file <Script package file path>

Uploads the script package to the parameter collection definition for the parameter collection definition ID. The input file is a script package zip file.

Script packages that can be uploaded are subject to the following conditions. An error will occur if conditions other than these are used.

- File extension must be .zip.

Ensure that the file extension is .zip.

- File size must be no greater than 2 MB.

The maximum uploadable file size is 2 MB.

- Store the discovery script in the root.

Store the discovery script (discover.cmd or discover.sh) in the root of the zip file rather than under the directory.

-download -id <Parameter collection definition ID> -dir <Output directory path>

Downloads the script package registered in the parameter collection definition of the parameter collection definition ID. Outputs script package zip files to the output directory path.

Ensure that no files exist in the output directory, or an error will occur.

-e <Character encoding>

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return value

Return value	Meaning
0	Completed normally.
10	A parameter error has occurred.
20	The specified I/O path is incorrect.
30	The specified character encoding is incorrect.
40	Unable to analyze the file.
50	The file description is incorrect.
70	The specified ID does not exist.
71	The specified ID is product-predefined.
72	The package has not been registered in the specified ID.
73	The package has not been correctly registered in the specified ID.
100	Systemwalker Software Configuration Manager is not running.

Return value	Meaning
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
120	There is not enough free disk space for the media library.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

```
<Systemwalker Software Configuration Manager installation directory>\SWCFMGM\bin
```

[Linux]

```
/opt/FJSVcfmgm/bin
```

Privilege required/execution environment

[Windows]

- Administrator privileges are required. Execute this command as an administrator.
- This command can be executed on the admin server.

[Linux]

- This command can only be executed by the system administrator (superuser).
- This command can be executed on the admin server.

Notes

Common notes

- This command cannot be executed simultaneously with other commands. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.

Notes regarding update

- The method cannot be changed if the software and a parameter collection definition are associated or a script package is registered.
- If multiple parameter collection definitions are associated with a single software product, you cannot specify a duplicate key for the parameter list.

Notes regarding deletion

- Use this command to delete information about parameter collection definitions that no longer need to be managed by Systemwalker Software Configuration Manager.

Notes regarding list display

- If no parameter collection definitions have been registered, only the header information will be output.

```
#Definition ID,Product predefined,Definition name,Method,Package name
```

Notes regarding additions and update errors

- If a parameter collection definition file is invalid, the registration and update of the parameter collection definitions will be canceled and the following error message will be output:

```
An error occurred during XML file validation.[Details:<Cause>]
```

Examples

- Add

[Windows]

```
swcfmg_param_collectingdef.exe -add -file C:\work\paramcollectingdef.xml
```

[Linux]

```
swcfmg_param_collectingdef -add -file /tmp/paramcollectingdef.xml
```

- Update

[Windows]

```
swcfmg_param_collectingdef.exe -update -id SC00000001 -file C:\work\paramcollectingdef.xml
```

[Linux]

```
swcfmg_param_collectingdef -update -id SC00000001 -file /tmp/paramcollectingdef.xml
```

- Delete

[Windows]

```
swcfmg_param_collectingdef.exe -delete -id SC00000001
```

[Linux]

```
swcfmg_param_collectingdef -delete -id SC00000001
```

- List

[Windows]

```
swcfmg_param_collectingdef.exe -list
```

[Linux]

```
swcfmg_param_collectingdef -list
```

- Acquire

[Windows]

```
swcfmg_param_collectingdef.exe -get -id SC00000001 -file C:\work\paramcollectingdef.xml
```

[Linux]

```
swcfmg_param_collectingdef -get -id SC00000001 -file /tmp/paramcollectingdef.xml
```

- Upload

[Windows]

```
swcfmg_param_collectingdef.exe -upload -id SC00000001 -file C:\work\paramcollectingdef.zip
```

[Linux]

```
swcfmg_param_collectingdef -upload -id SC00000001 -file /tmp/paramcollectingdef.zip
```

- Download

[Windows]

```
swcfmg_param_collectingdef.exe -download -id SC00000001 -dir C:\work\paramcollectingdef
```

[Linux]

```
swcfmg_param_collectingdef -download -id SC00000001 -dir /tmp/paramcollectingdef
```

5.5.3 swcfmg_param_defassoc (Command to Associate Software and Parameter Definitions)

Description

Associates or disassociates the software settings definition with the parameter settings definition, or the software collection definition with the parameter collection definition, and displays them in list form.

Synopsis

[Windows]

```
swcfmg_param_defassoc.exe  
{ -attach -softid <Software ID> -sdefid <Parameter settings definition ID> |  
-attach -softid <Software ID> -cdefid <Parameter collection definition ID> |  
-detach -softid <Software ID> -sdefid <Parameter settings definition ID> |  
-detach -softid <Software ID> -cdefid <Parameter collection definition ID> |  
-list [-sdef] [-cdef] [-file <Output file path>] [-e <Input file character encoding>] }
```

[Linux]

```
swcfmg_param_defassoc  
{ -attach -softid <Software ID> -sdefid <Parameter settings definition ID> |  
-attach -softid <Software ID> -cdefid <Parameter collection definition ID> |  
-detach -softid <Software ID> -sdefid <Parameter settings definition ID> |  
-detach -softid <Software ID> -cdefid <Parameter collection definition ID> |  
-list [-sdef] [-cdef] [-file <Output file path>] [-e <Input file character encoding>] }
```

Options

-attach -softid <Software ID> -sdefid <Parameter settings definition ID>

Associates the software that has the specified ID and the parameter settings definition that has the specified ID. Note that multiple parameter settings definitions can be associated with a single software product. A parameter definition method compatible with the OS type of the software must be specified.

-attach -softid <Software ID> -cdefid <Parameter collection definition ID>

Associates the software that has the specified ID and the parameter collection definition that has the specified ID. Note that a single parameter collection definition can be associated with multiple software products. A parameter definition method compatible with the OS type of the software must be specified.

-detach -softid <Software ID> -sdefid <Parameter settings definition ID>

Removes the association between the software that has the specified ID and the parameter settings definition that has the specified ID

-detach -softid <Software ID> -cdefid <Parameter collection definition ID>

Removes the association between the software that has the specified ID and the parameter collection definition that has the specified ID

-list [-sdef] [-cdef] [-file <Output file path>]

Outputs a list of associations in CSV format

-sdef Outputs a list of only those associations between the software and parameter settings definitions.

-cdef Outputs a list of only those associations between the software and parameter collection definitions.

If the specified file does not exist, a new file will be created. If the file exists, the existing file will be overwritten.

If output to the file fails due to an error, no new file will be created. If an existing file was to be overwritten, it will be kept as is without being overwritten.

- Output format for standard output

Outputs the list of associations in the following format:

Item No.	Item	Remarks
1	Software ID	Outputs the software ID.
2	Definition ID	Outputs the parameter settings ID or the parameter collection ID.
3	Software name	Outputs the software product name.
4	Version	Outputs the software version.
5	OS type	Outputs the software OS.
6	Definition name	Outputs the parameter settings definition name or the parameter collection definition name.

- Header format

Insert the header in the first line of the CSV file using the format shown below. The header is output even if there is no information to be output to the file.

#Software ID,Definition ID,Software name,Version,Vendor,Definition name

-e <Character encoding>

Specify the character encoding for the file.

The character encodings below can be specified (single-byte uppercase characters must be used):

- UTF-8

If this option is omitted, the character encoding of the execution environment of this command will be used.

Return value

Return value	Meaning
0	Completed normally.

Return value	Meaning
10	A parameter error has occurred.
20	The specified I/O path is incorrect.
30	The specified character encoding is incorrect.
70	The specified ID does not exist.
71	The software and parameter definition are already associated.
100	Systemwalker Software Configuration Manager is not running.
101	A command that cannot be executed at the same time as this command is executing.
102	You do not have the privileges required to execute this command.
110	A write error has occurred.
200	Setup has not been performed.
255	A system error has occurred.

Command location

Admin server

[Windows]

```
<Systemwalker Software Configuration Manager installation directory>\SWCFMGM\bin
```

[Linux]

```
/opt/FJSCvcmgm/bin
```

Privilege required/execution environment

[Windows]

- Administrator privileges are required. Execute this command as an administrator.
- This command can be executed on the admin server.

[Linux]

- This command can only be executed by the system administrator (superuser).
- This command can be executed on the admin server.

Notes

Common notes

- This command cannot be executed simultaneously with other commands. An exclusive control error will occur if a command is executed while another command is still executing.
- Execute this command while Systemwalker Software Configuration Manager is running.

Notes regarding association

- If multiple parameter collection definitions are associated with a single software product, you cannot specify a duplicate key for the parameter list.
- When the software type is Solaris, that software cannot be associated with the parameter definitions.

Notes regarding removal of associations

- Use this command to remove associations that no longer need to be managed by Systemwalker Software Configuration Manager.

Notes regarding list display

- If no predefined parameters have been registered, only the header information will be output.

```
#Software ID,Definition ID,Software name,Version,Vendor,Definition name
```

Examples

- Associate

[Windows]

```
swcfmg_param_defassoc.exe -attach -softid UDP00001 -sdefid SS00000001
```

[Linux]

```
swcfmg_param_defassoc -attach -softid UDP00001 -sdefid SS00000001
```

- Disassociate

[Windows]

```
swcfmg_param_defassoc.exe -detach -softid UDP00001 -sdefid SS00000001
```

[Linux]

```
swcfmg_param_defassoc -detach -softid UDP00001 -sdefid SS00000001
```

- List

[Windows]

```
swcfmg_param_defassoc.exe -list
```

[Linux]

```
swcfmg_param_defassoc -list
```

Chapter 6 File Reference

6.1 File List

The following table lists the definition files.

Type	Feature name	Overview
6.2 Definition files of the information collection source	Information Collection Source File	This file is used to configure information collection sources managed by Systemwalker Software Configuration Manager.
	Collection Item Information File	This file is used to configure collection items of information collection source managed by Systemwalker Software Configuration Manager.
	Collection Information CSV File	This file is used to define the information of the software installed on a server and the applied patch information.
6.3 Software Information Definition Files	Software Information File	Registers and updates software information.
	Installed Software Information File	Registers and deletes installed software information.
6.4 Definition files of the patch information	Patch Information File	This file is used to configure the patch information.
	Applied Patch Information File	This file is used to perform registration and deletion of applied patch information.
6.5 Parameter Information Definition Files	Parameter Settings Definition File	Defines configuration information for parameters that can be set in the software.
	Parameter Information File	Defines the values to be set in the software parameters.
	Parameter Collection Definition File	Defines configuration information for parameters to be collected by the software.

6.2 Definition files of the information collection source

This section explains the definition files of the information collection source.

6.2.1 Information Collection Source File

File name

Any name

Description

A file to manage information collection sources.

File location

Any folder

File format

CSV format

```
[#]<source-id>,<enable>  
...
```

Parameters

The following is regarded as a single definition:

[#]<source-id>,<enable>

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	source-id	Information collection source ID	Example: UpdateAdvisor	Cannot be omitted. Cannot be modified.
2	enable	Enabled/disabled	Example: true	Configure whether to collect information from the corresponding information collection source during discovery. When enabled, information is collected. When disabled, information is not collected. Manually registered information collection sources cannot be enabled. - true : enabled - false : disabled

Notes

- When installing this product, the following information collection sources are defined in advance.

```
Direct, false  
UpdateAdvisor, true
```

Examples

- During batch update

```
SourceDef, false
```

6.2.2 Collection Item Information File

File name

Any name

Description

A file to define the collection items of software and patches.

Register and update the collection items using the options of swcfmg_source_def (Information Collection Source Management Command).

File location

Any folder

File format

CSV format

```
[#]<number>,<collection-item>,<identification-item>,<definition-item>  
...
```


Parameters

The following is regarded as a single definition:

[#]<number>,<collection-item>,<identification-item>,<definition-item>

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	number	Order	Example: 1	Cannot be omitted. Cannot be modified. Specify a serial number starting from 1.
2	collection-item	Collection item	Example: softwareName	Cannot be omitted. Can be modified. For specifiable values, refer to " 1.1.2 List of Collection Items ".
3	identification-item	Identification item	Example: true	Cannot be omitted. Cannot be modified. Specification depends on whether the item is an identification item. When this collection item is an identification item, specify "true". When this collection item is not an identification item, specify "false".
4	definition-item	Definition item	Example: true	Cannot be omitted. Can be modified. Specification depends on whether the item is a definition item. When this collection item is a definition item, specify "true". When this collection item is not a definition item, specify "false". An identification item must be included in a definition item.

Notes

- When installing a product, the following collection items are defined in advance.

- UpdateAdvisor

- Software

```
1,softwareName,true,true
2,version,true,true
3,osType,true,true
4,vendor,false,true
```

- Patch

```
1,softwareName,true,true
2,osType,true,true
3,software,false,true
4,softwareName,false,true
5,significance,false,true
6,updateFormat,false,true
7,releaseDate,false,true
```

- Manual registration

- Software

```
1,softwareName,true,true
2,version,true,true
3,osType,true,true
4,vendor,false,true
```

Examples

- During registration

- Software

```
1,softwareName,true,true
2,version,true,true
3,osType,true,true
4,vendor,false,true
```

- Patch

```
1,patchName,true,true
2,osType,true,true
3,software,false,true
4,applyDate,false,false
```

- During update

- Software

```
1,softwareName,true,true
2,version,true,true
3,osType,true,true
4,vendor,false,true
5,installDate,false,false
```

- Patch

```
1,patchName,true,true
2,osType,true,true
3,software,false,true
4,applyDate,false,false
5,significance,false,true
```

6.2.3 Collection Information CSV File

File name

```
softwareinfo.csv or patchinfo.csv
```

Description

A file to output the information of the software installed in a server or the applied patch information. This file is transferred to the admin server, and the described software information or patch information are registered in CMDB as installed software or applied patches. Use this file in the following cases:

- When the information of software on a server in which discovery scripts are installed is output to the software information CSV file during the discovery of software patch information
- When the information of patches which are applied to a server is output to the patch information CSV file during the discovery of software patch information using discovery scripts

File location

Any folder

File format

CSV format

```
[#]<local-id>,<collection -info>,...  
...
```

Parameters

The following is regarded as a single definition:

```
[#]<local-id>,<collection-info>,...
```

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	local-id	Local ID	Example: software1	An ID temporarily used when registering the information described in this CSV file in CMDB. Specify a character string containing up to 32 alphanumeric characters, hyphens (-), and underscores (_). For the local ID, specify a unique value in this CSV file. A local ID cannot be registered in CMDB.
2	collection-info	The value of the first collection item	Example: SoftwareA	According to the definition of the collection item, specify the values of the collection items in the defined order. The collection items for software and patches are different depending on the information collection source. Display details using swcfmg_source_def (Information Collection Source Management Command) to check the collection items. For details on the specifiable values for each collection item, refer to " 1.1.2 List of Collection Items ".
3		The value of the second collection item	Example: V1.0	
...		

Notes

None

Examples

None

6.3 Software Information Definition Files

This section explains the software information definition files.

6.3.1 Software Information File

File name

Any name

Description

This file is used to register and update software information.

Register and update software information using the `swcfmg_software` (Software Information Management Command) options.

File location

Any folder

File format

CSV format

```
[#][<software-id>],software-info,..
...
```

When installing this product, the software whose information collection sources were manually registered is in the following CSV format.

```
[#][<software-id>],<software-name>,<version>,<os type>,<vendor>
...
```

Parameters

The following is regarded as a single definition:

```
[#][<software-id>],<software-info>,...
```

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	software-id	Software ID	Example: 4TIT42JIF @Direct	Can be omitted during registration. Cannot be omitted during updates. Cannot be modified. For registration, use the following format: An arbitrary string beginning with an alphanumeric character and containing up to 32 alphanumeric characters, hyphens (-), and underscores (_). <i>Arbitrary character string + "@" + information collection source ID</i> When omitted during registration, the ID is automatically assigned. A software ID which is already in use cannot be used. Specify the ID output in the list displayed during update.
2	software-info	The value of the first collection item	Example: ABCDE-Software	According to the definition of the software collection item, specify the values of the collection items in the defined order.
3		The value of the second collection item	Example: 1.0	

Column	Item	Meaning	Value	Remarks
...		Use swcfmg_source_def (Information Collection Source Management Command) to check the defined collection item. For details on the specifiable value for each collection item, refer to " 1.1.2 List of Collection Items ".

When installing this product, if the information collection source is manually registered software, the following is regarded as a single definition:

[#][<software-id>,<software-name>,<version>,<os-type>,<vendor>

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	software-id	Software ID	Example: 4TIT42JIF @Direct	Can be omitted during registration. Cannot be omitted during updates. Cannot be modified. For registration, use the following format: An arbitrary string beginning with an alphanumeric character and containing up to 32 alphanumeric characters, hyphens (-), and underscores (_). <i>Arbitrary character string + "@Direct"</i> When omitted during registration, the ID is automatically assigned. A software ID which is already in use cannot be used. Specify the ID output in the list displayed during update.
2	software-name	Software name	Example: ABCDE- Software	Cannot be omitted. Cannot be modified.
3	version	Software version	Example: 1.0	Cannot be omitted. Cannot be modified.
4	os type	OS type	Example: Windows	Cannot be omitted. Cannot be modified. The following types can be specified: - Windows - Windows EM64T - Linux - Linux EM64T - Solaris
5	vendor	Software vendor name	Example: ABCDE Company	Can be omitted. Can be modified.

Notes

None

Examples

- During registration

```
,ABCDE-Software,1.0,Windows,ABCDE Company  
,FGHIJ-Software,2.0,Windows, ABCDE Company
```

- During update

```
L31CWYNJI@Direct,ABCDE-Software,1.1,Windows,ABCDE Company  
2KITGRJV8@Direct,FGHIJ-Software,2.0.1,Windows, ABCDE Company
```

6.3.2 Installed Software Information File

File name

Any name

Description

This file is used to register and delete installed software information.

Register and delete installed software information using the `swcimg_installedsoftware` (Installed Software Information Management Command) options.

File location

Any folder

File format

CSV format

```
[#]<server>,<software-id>
```

...

Parameters

The following constitutes a single definition:

```
[#]<server>,<software-id>
```

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	server	Management IP address or server ID of the managed server	Example: 10.10.10.1 0	Cannot be omitted.
2	software-id	Software ID	Example: UDP00001	Cannot be omitted. When specifying a multi-instance installed software, specify the software IDs and installation locations separated by semicolons (";"). <i>Software ID + ";" + installation location</i>

Notes

None

Examples

```
10.10.10.11, L31CWYNJI@Direct
10.10.10.11, 2KITGRJV8@Direct
10.10.10.12, L31CWYNJI@Direct
```

6.4 Definition files of the patch information

This section explains the patch information files.

6.4.1 Patch Information File

File name

Any name

Description

A file to perform registration and update of patch information.

Register and update patch information using the options of `swcfmg_patch_def` (Patch Information Management Command).

File location

Any folder

File format

CSV format

```
[#][<patch-id>],patch-info,...
...
```

Parameters

The following is regarded as a single definition:

```
[#][<patch-id>],patch-info,...
```

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	patch-id	Patch ID	Example: GEWFSGR 42@Direct	Can be omitted during registration. Cannot be omitted during updates. Cannot be modified. For registration, use the following format: An arbitrary string beginning with an alphanumeric character and containing up to 32 alphanumeric characters, hyphens (-), and underscores (_). <i>Arbitrary character string + "@" + information collection source ID</i> When omitted during registration, the ID is automatically assigned.

Column	Item	Meaning	Value	Remarks
				A patch ID which is already in use cannot be used. Specify the ID output in the list displayed during update.
2	patch-info	The value of the first collection item	Example: 3811942	According to the definition of the patch collection item, specify the values of the collection items in the defined order. Use swcfmg_source_def (Information Collection Source Management Command) to check the defined collection item. For details on the specifiable value for each collection item, refer to " 1.1.2 List of Collection Items ".
3		The value of the second collection item	Example: 9.3	
...		

Notes

None

Examples

- During registration

```
,PatchA,Windows,L31CWYNJI@Direct,1
```

- During update

```
9G4G20GWD@Direct,PatchA,Windows,L31CWYNJI@Direct,2
```

6.4.2 Applied Patch Information File

File name

Any name

Description

A file to perform registration and deletion of applied patch information.

Register and delete applied patch information using the options of swcfmg_patch_applied (Applied Patch Information Management Command).

File location

Any folder

File format

CSV format

```
[#]<server>,<software-id>,<patch-id>  
...
```

Parameters

The following is regarded as a single definition:

[#]<server>,<software-id>,<patch-id>

Column	Item	Meaning	Value	Remarks
	#	Comment line		
1	server	IP address or server ID of the managed server	Example: 10.10.10.10	Cannot be omitted.
2	software-id	Software ID	Example: UDP00001	Cannot be omitted. When specifying a multi-instance installed software, specify the software IDs and installation locations separated by semicolons (";"). <i>Software ID + ";" + installation location</i>
3	patch-id	Patch ID	Example:	Cannot be omitted.

Notes

None

Examples

```
10.10.10.11,L31CWYNJI@Direct,GYW7RTII3@Direct  
10.10.10.11,2KITGRJV8@Direct,HZX8TC184@Direct  
10.10.10.12,L31CWYNJI@Direct,GYW7RTII3@Direct
```

6.5 Parameter Information Definition Files

This section explains the parameter information definition files.

6.5.1 Parameter Settings Definition File

File name

Any time

Description

This is an XML file that describes the configuration information for parameters that can be set in the software.

Refer to "[4.1 Definition of Parameters to be Set](#)" for information on parameter settings definitions.

File format

Parameter settings definition XML files will have the following format:

```
<?xml version="1.0" encoding="UTF-8"?>  
<parameterSetting version="1.0">  
  <name>parmSettingsDefinitionName</name>  
  <description>desc</description>  
  <method>parmSetupMethod</method>  
  <parameters>  
    <parameter>  
      <key>[parmKey]</key>  
      <type>[parmValueType]</type>  
      <value>[defaultValIfParmIsMandatory (if type is boolean, number, or string)]</value>  
      <array>[defaultValIfParmIsMandatory (if type is string array)]  
        <element>elementVal</element>
```

```

    ...
  </array>
  <map>[ default tVal / fParm / sMandatory> (if type is map)]
    <entry>
      <subkey>mapSubkey</subkey>
      <value>mapVal</value>
    </entry>
    ...
  </map>
  <label>[ parmLabel ]</label>
  <description>[ parmDesc]</description>
</parameter>
...
</parameters>
</parameterSetting>

```

The table below describes the items (tags) and their settings.

Modify parameter settings definitions as necessary, based on the information in this table.

Tag names in square brackets [] are optional.

Tag name	Type	Allowable range	Description	Mandatory	Settings
name	string UTF-8	256 characters or less	Specifies the parameter settings definition name.	O	
description	string UTF-8	256 characters or less	Specifies the parameter settings definition description.	O	
method	string ASCII	Select an option	Specifies the parameter setup method.	Y	Select from the following options: - "cmd": Calls the startup script (startup.cmd). - "sh": Calls the startup script (startup.sh).
parameters	-	-	Specifies multiple parameters that can be configured in the software.	N	
parameter	-	1 or more	Specifies the parameter key and value pairs that can be configured in the software.	N	
key	string ASCII	256 bytes or less	Specifies the parameter key.	Y	Characters that can be used are alphanumeric characters, ".", "_", and "-". However, the first character must only be alphabetic.
type	string ASCII	Select an option	Specifies the value type.	Y	Select from the following options:

Tag name	Type	Allowable range	Description	Mandatory	Settings
					<ul style="list-style-type: none"> - boolean: true, false - number: Number - string: Character string - string array: Character string array - map Refer to " 4.1.2 Parameter Types " for information on types.
[value]	string UTF-8	4096 characters or less	<p>Specifies the default value if a value must be set. The value can be changed when configuring this parameter, but it cannot be left blank.</p> <p>Can only be specified if type is boolean, number or string.</p> <p>The value, array, and map elements are mutually exclusive.</p>	O	The <type> tag determines what type of value can be specified. Refer to " 4.1 Definition of Parameters to be Set " for information on values.
[array]	-	-	<p>Specifies the default value if a value must be set.</p> <p>Can only be specified if the type is string array.</p> <p>The value, array, and map elements are mutually exclusive.</p>	N	Use the element tag to specify array elements. Do not specify the element tag if there are zero arrays.
[element]	string UTF-8	4096 characters or less	<p>Specifies the elements in an array when the default value type for the mandatory parameter is "array".</p>	Y	Use a character string to specify the elements in an array. It is also possible to specify an empty string.
[map]	-	-	<p>Specifies the default value if a value must be set.</p> <p>Can only be specified if type is map.</p> <p>. The value, array, and map elements are mutually exclusive.</p>	N	Use the entry tag to specify map entries. Do not specify the entry tag if there are zero maps.
[entry]	-	-	Used with each key-value pair to specify	N	Use the key tag to specify keys and the

Tag name	Type	Allowable range	Description	Mandatory	Settings
			the value when the default value type for the mandatory parameter is map.		value tag to specify values.
subkey	string ASCII	256 bytes or less	Specifies the subkey when the default value type for the mandatory parameter is map.	Y	Characters that can be used are alphanumeric characters, periods (.), underscores (_), and hyphens (-). However, the first character must only be alphabetic.
value	string UTF-8	4096 characters or less	Specifies the value when the default value type for the mandatory parameter is map.	Y	Use a character string to specify map values. It is also possible to specify an empty string.
[label]	string UTF-8	64 characters or less	Specifies the label used to display the parameter in the window.	O	
[description]	string UTF-8	256 characters or less	Specifies the parameter description.	O	

Legend for column 'Mandatory':

Y: If the tag is specified, then a value is also required

O: Optional

N: The tag can be specified without a value

Example (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<parameterSetting version="1.0">
  <name>Setting Definition</name>
  <description>Parameter Setting Definition</description>
  <method>cmd</method>
  <parameters>
    <parameter>
      <key>key.isParameter</key>
      <type>boolean</type>
      <value>>true</value>
      <label>isParameter</label>
      <description>boolean</description>
    </parameter>
    <parameter>
      <key>key.number</key>
      <type>number</type>
      <value>100</value>
      <label>number</label>
      <description>number</description>
    </parameter>
    <parameter>
      <key>key.parameter</key>
      <type>string</type>
    </parameter>
  </parameters>
</parameterSetting>
```

```

        <value>parameter</value>
        <label>parameter</label>
        <description>string</description>
    </parameter>
    <parameter>
        <key>key.parameter.list</key>
        <type>string array</type>
        <array>
            <element>first</element>
            <element>second</element>
        </array>
        <label>list</label>
        <description>array</description>
    </parameter>
    <parameter>
        <key>key.parameters</key>
        <type>map</type>
        <map>
            <entry>
                <subkey>subkey.right</subkey>
                <value>right</value>
            </entry>
            <entry>
                <subkey>subkey.left</subkey>
                <value>left</value>
            </entry>
        </map>
        <label>parameters</label>
        <description>map</description>
    </parameter>
</parameters>
</parameterSetting>

```

6.5.2 Parameter Information File

File name

Any name

Description

This is a file in CSV or XML format that describes the values to be set in the software parameters. Parameters for which values can be set in the software are the parameters defined in the parameter settings definition.

Refer to "Parameter Value Settings" in the *Operation Guide* for details on parameter information.

CSV file format

Parameter information CSV files will have the format shown below.

Modify the parameter information as necessary, based on this information.

Purpose of line	Line/row	Row 1	Row 2	Row 3	Row 4
Header	Line 1	#Parameters	Key	Value	Status
Name	Line 2	parameterInfo	name	[Parameter information name]	
Description	Line 3	parameterInfo	description	[Parameter information description]	

Purpose of line	Line/row	Row 1	Row 2	Row 3	Row 4
Parameters	Line 4	parameters	[Parameter key]	[Parameter value]	[Parameter status]
...	Line 5...	parameters

Explanation of lines

The following table explains the lines and their settings:

Purpose of line	Description	Mandatory	Settings
Header	The header is deemed to be specified if the first line of the CSV file starts with a hash (#) symbol.	N	Apart from the # symbol, the header may contain any content. Omit the line if not specifying a header.
Name	This line specifies the parameter information name.	Y	
Description	This line specifies the parameter information description.	N	Omit the line if not specifying a parameter information description.
Parameters	This line specifies the parameter key and value.	N	Add the same number of lines as the number of parameter keys you wish to specify. Omit the line if not specifying any parameter keys.

Explanation of items

The following table explains the item names and their settings.

Item names in square brackets [] can be omitted line by line. Omit the fourth row if omitting all parameter statuses.

Item	Type	Allowable range	Description	Mandatory	Settings
Parameter information name	string	256 characters or less	Specifies the parameter information name.	N	
[Parameter information description]	string	256 characters or less	Specifies the description of the parameter information.	N	
Parameter key	string	If type is boolean, number or string: 256 bytes or less If type is string array or map: 514 bytes or less	If type is boolean, number of string: specifies the parameter key. If type is string array: specifies the element index. If type is map: specifies the subkey.	Y	Only the key contained in the parameter settings definition can be specified. Enclose the string array index and the map subkey in square brackets ([]). The string array index starts from 1. When using a parameter settings definition from "Parameter Reference", specify the parameter key.
Parameter value	string	4096 characters or less	Specifies the parameter value.	N	The value depends on the parameter settings definition restrictions.

Item	Type	Allowable range	Description	Mandatory	Settings
					Refer to "Predefined Parameters" of the <i>Operation Guide</i> for information on values.
Parameter status	string	32 characters or less	Specifies the parameter value.	N	Specify whether to set a parameter or not. The following values can be set to indicate the status: <ul style="list-style-type: none"> - Set a value: set - Do not set a value: Empty string, notset

Key-value description

Keys and values should be specified as shown in the table below:

Parameters		Specification	
Type	Value characteristics	Key	Value
boolean	Value specified	Key	Value
number	Value specified	Key	Value
string	Value specified	Key	Value
	Empty string	Key	Empty string
string array	Value specified (1 or more elements)	Key [index]	Value
	Empty string (1 or more elements)	Key [index]	Empty string
	0 elements	Key	Empty string
map	Value specified (1 or more entries)	Key [subkey]	Value
	Empty string (1 or more entries)	Key [subkey]	Empty string
	0 entries	Key	Empty string

Example (CSV)

```
#Parameters,Key,Value,Status
parameterInfo,name,Parameter Info,
parameterInfo,description,Parameter Info,
parameters,key.isParameter,true,set
parameters,key.number,100,set
parameters,key.parameter,parameter,set
parameters,key.parameter.list[1],first,set
parameters,key.parameter.list[2],second,set
parameters,key.parameter.list[subkey.right],right,set
parameters,key.parameter.list[subkey.left],left,set
```

XML file format

Parameter information XML files will have the format shown below.

Modify the parameter information as necessary, based on this information.

```

<?xml version="1.0" encoding="UTF-8"?>
<parameterInfo version="3.0">
  <name><parm information name></name>
  <description><desc></description>
  <parameters>
    <parameter>
      <key><parm key></key>
      <value><parm val> (if type is boolean, number, or string)</value>
      <array><parm val> (if type is string array)
        <element><element val> </element>
        ...
      </array>
      <map><parm val> (if type is map) *1 *4
        <entry>
          <subkey><map subkey></subkey>
          <value><map val></value>
        </entry>
        ...
      </map>
    </parameter>
    ...
  </parameters>
</parameterInfo>

```

The table below describes the items (tags) and their settings.

Tag names in square brackets [] are optional.

Tag name	Type	Allowable range	Description	Mandatory	Settings
name	string UTF-8	256 characters or less	Specifies the parameter information name.	O	
[description]	string UTF-8	256 characters or less	Specifies the description of the parameter information.	O	
parameters	-	-	Specifies multiple parameters to be set in the software.	N	
[parameter]	-	0 or more	Specifies the parameter key and value pairs to be set in the software.	N	
key	string ASCII	1 to 256 bytes	Specifies the parameter key.	Y	Only the key contained in the parameter settings definition can be specified. When using a parameter settings definition from "Parameter Reference", specify the parameter key.
[value]	string UTF-8	4096 characters or less	Specifies the parameter value. Can only be specified if type is boolean, number or string.	O	The value depends on the parameter settings definition restrictions. Refer to "Predefined Parameters" of <i>Operation Guide</i> for

Tag name	Type	Allowable range	Description	Mandatory	Settings
			One and only one <value>, <array>, or <map> tag must be specified.		information on values.
[array]	-	-	Specifies the parameter value. Can only be specified if type is string array. One and only one <value>, <array>, or <map> tag must be specified.	N	Use the element tag to specify array elements. Do not specify the element tag if there are zero arrays.
[element]	string UTF-8	4096 characters or less	Specifies the elements in an array when the parameter value type is "array".	Y	Use a character string to specify the elements in an array. It is also possible to specify an empty string.
[map]	-	-	Specifies the parameter value. Can only be specified if type is map. One and only one <value>, <array>, or <map> tag must be specified.	N	Use the entry tag to specify map entries. Do not specify the entry tag if there are zero maps.
[entry]	-	-	Used with each key-value pair to specify the value when the parameter value type is map.	N	Use the key tag to specify keys and the value tag to specify values.
subkey	string ASCII	1 to 256 bytes	Specifies the subkey when the parameter value type is map.	Y	Characters that can be used are alphanumeric characters, periods (.), underscores (_), and hyphens (-). However, the first character must only be alphabetic.
value	string UTF-8	4096 characters or less	Specifies the value when the parameter value type is map.	Y	Use a character string to specify map values. It is also possible to specify an empty string.

Legend for column 'Mandatory':

Y: If the tag is specified, then a value is also required

O: Optional

N: The tag can be specified without a value

Example (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<parameterInfo version="3.0">
```

```

<name>Parameter Info</name>
<description>Parameter Info</description>
<parameters>
  <parameter>
    <key>key.isParameter</key>
    <value>true</value>
  </parameter>
  <parameter>
    <key>key.number</key>
    <value>100</value>
  </parameter>
  <parameter>
    <key>key.parameter</key>
    <value>parameter</value>
  </parameter>
  <parameter>
    <key>key.parameter.list</key>
    <array>
      <element>first</element>
      <element>second</element>
    </array>
  </parameter>
  <parameter>
    <key>key.parameters</key>
    <map>
      <entry>
        <subkey>subkey.right</subkey>
        <value>right</value>
      </entry>
      <entry>
        <subkey>subkey.left</subkey>
        <value>left</value>
      </entry>
    </map>
  </parameter>
</parameters>
</parameterInfo>

```

6.5.3 Parameter Collection Definition File

File name

Any name

Description

This is an XML file that describes the configuration information for parameters to be collected by the software.

Refer to "[4.2 Definition of Parameters to be Collected](#)" for information on parameter collection definitions.

File format

Parameter collection definition XML files will have the following format:

```

<?xml version="1.0" encoding="UTF-8"?>
<parameterCollecting version="1.0">
  <name>parameterCollectionName</name>
  <method>parameterCollectionMethod</method>
  <parameters>
    <parameter>
      <key>[parameterKey]</key>
      <type>[parameterValueType]</type>
      <label>[parameterLabel]</label>
    </parameter>
  </parameters>
</parameterCollecting>

```

```

        </parameter>
        ...
    </parameters>
</parameterCollecting>

```

The table below describes the items (tags) and their settings.

Modify parameter settings definitions as necessary, based on the information in this table.

Tag names in square brackets [] are optional.

Tag name	Type	Allowable range	Description	Mandatory	Settings
name	string UTF-8	256 characters or less	Specifies the parameter collection name.	O	
method	string ASCII	Select an option	Specifies the parameter collection method.	Y	Select from the following options: <ul style="list-style-type: none"> - "cmd": Calls the discovery script (discover.cmd) - "sh": Calls the discovery script (discover.sh)
parameters	-	-	Specifies multiple parameters that can be collected from the software.	N	
parameter	-	1 or more	Specifies the parameter key and value pairs that can be collected from the software.	N	
key	string ASCII	1 to 256 bytes	Specifies the parameter key.	Y	Characters that can be used are alphanumeric characters, ".", "_", and "-". However, the first character must only be alphabetic.
type	string ASCII	Select an option	Specifies the value type.	Y	Select from the following options: <ul style="list-style-type: none"> - boolean - number - string - string array - map Refer to " 4.1.2 Parameter Types " for information on types.
[label]	string UTF-8	64 characters or less	Specifies the label used to display the parameter in the window.	O	

Legend for column 'Mandatory':

Y: If the tag is specified, then a value is also required

O: Optional

N: The tag can be specified without a value

Example (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<parameterCollecting version="1.0">
  <name>Collecting Definition</name>
  <method>cmd</method>
  <parameters>
    <parameter>
      <key>key.isParameter</key>
      <type>boolean</type>
    </parameter>
    <parameter>
      <key>key.number</key>
      <type>number</type>
    </parameter>
    <parameter>
      <key>key.parameter</key>
      <type>string</type>
    </parameter>
    <parameter>
      <key>key.parameter.list</key>
      <type>string array</type>
    </parameter>
    <parameter>
      <key>key.parameters</key>
      <type>map</type>
    </parameter>
  </parameters>
</parameterCollecting>
```

Chapter 7 Script Reference

This chapter explains the scripts provided by Systemwalker Software Configuration Manager.

7.1 Script List

The following table lists the scripts used for managing parameters.

Type	Feature name	Overview
Software Patch Collection Script	Discovery Script	A script to output the information of the software installed on a server or the applied patch information to a file.
Parameter settings script	Startup Script	Configures the software parameters.
Parameter collection script	Discovery Script	Collects parameter information from the software and outputs it to a file.

7.2 Software Patch Collection Script

This section explains the software patch collection script.

7.2.1 Discovery Script

Synopsis

```
discover -dir "output dir"
```

Options

-dir

Specify the output directory for the software information CSV file (softwareinfo.csv) and the patch information CSV file (patchinfo.csv). If the path contains spaces, enclose it in double quotation marks (").

Output format

"0" is returned if the script ends normally, and some other value is returned otherwise - error messages are sent either to the standard output or to the standard error output.

```
+ "output directory"  
+ Software information CSV file of collected software (softwareinfo.csv)  
+ Patch information CSV file of collected patches (patchinfo.csv)
```

Notes

Common notes

- The directory in which the command is stored will be considered the current directory.
- An error will occur if the directory does not exist, or if it exists and contains files or directories.
- Output the software information CSV file and patch information CSV file using the character encoding of the system of the server on which this script is running.

Notes regarding output of the software information CSV file

- Describe a single entry of information of software installed on the server per line of CSV. For details on the CSV file, refer to "[6.2.3 Collection Information CSV File](#)".

- The same software cannot be described multiple times. When describing multi-instance software, specify a different value for the installation location.
- The value of the identification item cannot be omitted. When the value is omitted or incorrect, the software information in the line indicated will not be registered. The identification item of the OS type can be omitted. When omitted, the OS type of the target server is configured.
- The values of collection items other than identification items can be omitted. When the value is omitted or incorrect, the value specified in the corresponding software definition will be configured. When the value is not configured in the software definition, this collection item will not be collected.

Notes regarding output of the patch information CSV file

- Describe a single entry of information of patches applied to the server per line of CSV. For details on the CSV file, refer to "[6.2.3 Collection Information CSV File](#)".
- The same patch cannot be applied to the same installed software multiple times. A single patch cannot be applied to multiple software.
- Specify the software to which the patch is applied in the collection item of the target software. For the software information CSV file, which is simultaneously output, specify the local ID of the corresponding software. The collection item of the target software cannot be omitted.
- The value of the identification item cannot be omitted. When the value is omitted or incorrect, the patch information in the line indicated will not be registered. The identification item of the OS type can be omitted. When omitted, the OS type of the target server is configured.
- The values of collection items other than identification items can be omitted. When the value is omitted or incorrect, the value specified in the corresponding patch definition will be configured. When the value is not configured in the patch definition, this collection item will not be collected. When the collection item of the software name is omitted and a value is not configured in the patch definition, the name of the software to be patched is configured.

Output Example

Output examples of the software information CSV file and the patch information CSV file are shown below. When the collection items of software and patches are defined as follows, and when a single entry of software information installed on a server and information of patches applied to that software are collected, specify the software information CSV file and the patch information CSV file as follows.

	Order	Collection Item	Identification Item
Collection items of the software	1	Software name	Yes
	2	Version	Yes
	3	OS type	Yes
	4	Installation location	-
Patch collection item	1	Patch name	Yes
	2	OS type	Yes
	3	Target software	-
	4	Application date	-

Specification examples

- Software information CSV file

```
software1,SoftwareA,V1.0,,C:\work
```

- Patch information CSV file

```
patch1,PatchA,,software1,2015/3/4 14:14
```

7.3 Parameter Settings Script

This section explains the parameter settings script.

7.3.1 Startup Script

Synopsis

```
startup [-installpath <installation location>]
```

Description

Configures software parameters.

Options

-installpath

When the installation location is configured in the installed software, the installation location is specified. If the path contains spaces, enclose it in double quotation marks (""). When the installation location is not configured in the installed software, this option is omitted.

Output format

"0" is returned if the script ends normally, and some other value is returned otherwise - error messages are sent either to the standard output or to the standard error output.

Notes

The directory in which the command is stored will be considered the current directory.

The startup script location and the files that can be referenced by it are displayed below:

```
<work directory>
+ <software ID>
+ scriptpkg
| + Startup script(startup.cmd, startup.sh)
| + Environment variable setup script(setenv.cmd, setenv.sh)
| + Script package file attachments
+ paramdata
| +XML file with parameter information to be set(parameterinfo.xml)
+ parampkg
  + Parameter package file attachments
```

7.4 Parameter Collection Script

This section explains the parameter collection script.

7.4.1 Discovery Script

Synopsis

```
discover -dir <output dir> [-installpath installation location]
```

Description

Collects parameter information from the software and outputs it to a file.

Options

-dir

Specify the output directory for the parameter information XML file (parameterinfo.xml) and file attachments.

If the path contains spaces, enclose it in double quotation marks ("").

-installpath

When the installation location is configured in the installed software, the installation location is specified. To include a blank space in the path, enclose it with double quotes ("). When the installation location is not configured in the installed software, this option is omitted.

Output format

"0" is returned if the script ends normally, and some other value is returned otherwise - error messages are sent either to the standard output or to the standard error output. The output is sent to the directory below, in the format below.

```
+ <output directory>
+ XML file with information about parameters collected (parameterinfo.xml)
+ File attachment with collected parameters (includes those that will be output and those that will
not)
```

Notes

The directory which stores the command will be used as the current directory.

An error will occur if the directory does not exist, or if it exists and contains files or directories.

Chapter 8 Information Collection Source Reference

This chapter explains the information collection sources defined in advance in this product.

8.1 Information Collection Sources

This section provides a list of the information collection sources defined in advance.

Information Collection Source	Overview
Manual registration	Registers the information of software and patches which cannot be automatically collected during discovery.
UpdateAdvisor	Collects Fujitsu middleware information and Fujitsu middleware patch information.

8.2 Manual Registration

Use this function when managing the information of software and patches which cannot be automatically collected during discovery. Only the collection items of the software are defined. When managing patches, use the Information Collection Management Command to define the patch collection items. After this, use the Software Information Management Command and the Patch Information Management Command to define the software and patch information, and use the Installed Software Information Management Command and the Applied Information Management Command to register the installed software and the applied patches.

Collection items of the software

Collection Item	Identification Item	Definition Item
Software name	Yes	Yes
Version	Yes	Yes
OS type	Yes	Yes
Vendor	-	Yes

8.3 UpdateAdvisor

Coordinating with UpdateAdvisor (middleware) during discovery, collect the Fujitsu middleware information installed in a server and the applied Fujitsu middleware patch information. To collect the information, it is necessary to install UpdateAdvisor (middleware) in advance.

Software collection items

Collection Item	Identification Information	Definition Information
Software name	Yes	Yes
Version	Yes	Yes
OS type	Yes	Yes
Vendor	-	Yes

Patch collection items

Collection Item	Identification Information	Definition Information
Patch name	Yes	Yes
OS type	Yes	Yes

Collection Item	Identification Information	Definition Information
Software name	-	Yes
Importance	-	Yes